
AMSTRAD
CPC 464

SOFT158A

DDI-1 Firmware

An appendix to SOFT158, describing the CPC464 disc interface ROM routines and explanations

Paul Overell, Locomotive Software

目　录

附录　XIII

磁盘系统

第一章　概况……………………………………………… 1

1．1　cp/M2.2 ……………………………………… 2

1．2　AMSDOS ……………………………………… 4

1．3　实用程序……………………………………… 7

第二章、CP/M2.2和BIOS ……………………………… 11

2．1　cp/M的结构……………………………………… 11

2．2　存储器安排图……………………………………… 12

2．3　冷引导……………………………………………… 13

2．4　热引导……………………………………………… 14

2．5　驻留系统扩充……………………………………… 14

2．6　MOVCPM-COM的使用 ……………………… 15

2．7　IOBYTE。 ……………………………………… 16

2．8　BIOS用户间的相互作用。 ……………………… 18

2．9　初始命令缓冲器。………………………………… 19

2．10 BIOS的一些信息。 ……………………………… 19

2．11 磁盘的格式………………………………………… 21

2．12 引导扇区…………………………………………… 23

2.13 格式扇区…………………………………………………… 24
2.14 BIOS跳转块 ……………………………………………… 28
2.15 扩充的磁盘参数块………………………………………… 49
2.16 重启指令和它们的安排…………………………………… 53
2.17 替换寄存器组以及ENTER FIREWARE 程序的使用 ………………………………………………… 54

第三章 AMSDOS ……………………………………………… 56

3.1 特性…………………………………………………………… 56
3.2 文件名………………………………………………………… 58
3.3 文件头标……………………………………………………… 60
3.4 磁盘的更换…………………………………………………… 62
3.5 磁带固件程序的阻断………………………………………… 62
3.6 外部命令……………………………………………………… 78
3.7 AMSDOS的一些信息 ……………………………………… 85
3.8 可用于AMSDOS中的BIOS的一些特性 ……… 86
3.9 存储器的要求………………………………………………… 87

第四章 实用程序……………………………………………… 89

4.1 AMSDOS.COM …………………………………………… 89
4.2 CLOAD.COM……………………………………………… 90
4.3 CSAVE.COM ……………………………………………… 93
4.4 CHKDISC.COM …………………………………………… 95
4.5 DISCCHK.COM …………………………………………… 99
4.6 COPYDISC.COM ………………………………………… 104
4.7 DISCCOPY.COM ………………………………………… 109

4．8　FILECOPY．COM ………………………………… 114
4．9　FORMAT．COM …………………………………… 122
4．10 SETUP．COM ……………………………………… 126
4．11 BOOTGEN．COM和SYSGEN．COM ……… 147
第五章　硬件…………………………………………… 153
5．1　磁盘接口……………………………………… 153
5．2　串行接口……………………………………… 156

第一章　概　　况

磁盘系统的硬件包含了一个DDI-1接口和一个和两个 FD-1软盘驱动器。DDI-1接口插在CPC464机器背后标有“FLOPPY DISC”的插座上。磁盘驱动器通过一根电缆与该接口相连接。每个磁盘驱动器都用一个3”的软盘。磁盘的两面都可以使用，这取决于插入在驱动器中的磁盘本身。DDI-1 接口含有一个16k的扩充 ROM，其中的8K存放了磁盘驱动用的软件，其它剩下的部分由DRLOGO所使用。

这里提供了两个磁盘操作系统，一个是 AMSDOS，另一个是cp/м。AMSDOS可以使BASIC程序象使用盒式磁带文件一样来使用磁盘文件；cp/м2。2则是一个工业标准的操作系统。AMSDOS和cp/м都使用同样的文件结构并可以对彼此的文件进行读写。cp/м2.2通过打入｜CPM而从BASIC中进行调用。一部分cp/м（CCP和BDOS）通过驱动器A的磁盘而装入；cp/м BIOS驱留在DDI-1 ROM之中。

AMSDOS只要在BASIC与其所连接的DDI-1接口一起使用的时候便可启动。这截夺了大多数的盒式磁带的固件程序，并将它们更改到磁盘的固件程序。这样，只需作很少的修改或甚至于不必作什么修改就可能将正在使用磁带文件的BASIC程序变为使用磁盘文件。AMSDOS也提供了许多外部命令，它们可以删掉一些文件，更改一些文件的名称以及对磁带固件程序进行更改等等。

磁盘系统有许多用于进行格式化，复制磁盘和改变各种各样系统参数的实用程序。这些程序都是在cp/M下进行运行的。

1．1 CP/M2.2

cp/M2.2是适用于8080/8085/Z80的工业标准的操作系统，有了它，差不多可以使用成百个应用程序。

使用|CPM这条命令，可以通过BASIC来调用cp/M。

CPC464机器中的cp/M提供了一个39.5k的暂存程序区（TPA），每面180k容量的磁盘，IOBYTE，支持了一个两个通道的串行接口以及对低一级的设备进行访问的一些驱动程序。

TPA（暂存程序区）是RAM的一个区域，在这个区域中装有一个应用程序，并且可以进行运行，这个区域一般开始于＃0100地址。该区域的结束地址随不同的cp/M系统而有所不同。在TPA后面的首字节的地址以字的形式存放在地址＃0006中。在CPC464中，TPA一般扩充到＃9F05。（见第2章）。

磁盘具有每面180k的字节的容量，对cp/M系统来讲，保留了两条磁道，还有2k是用于目录所占用。因此，每一面所剩下的169k字节用于文件本身，对于特殊的使用，可采用另外两种磁盘的格式。（2．11节）。通过修改有关的扩充的磁盘参数块（XPB），则使用其它非AMSTRAD的磁盘格式也是可能的。（见2．15节）。

IOBYTE是一个可选的cp/M的特性，它可以用来更改字符的输入/输出。在CPC464中，完全提供有这种性能。关于IOBYIE的详细使用情况可以参阅cp/M操作系统的手册。

BIOS，如果它被设置的话，则可以用来驱动一个具有两

个通道的异步串行接口。这个接口不与DDI-1一起提供的。（见5.2节）。

对于一个应用程序来讲，通过一个扩充的BIOS跳转块，可以使用许多低级设备的驱动程序。这些程序包括对磁盘的扇区进行读写，对一道磁道进行格式化，指定各种各样的系统参数以及对串行接口进行初始化等等。（见2.14节）。

对于CP/M来讲，现有的一些应用程序的写入应该毫无困难地在CPC464中运行。可能有的一些问题只是：

· 使用Z80替换寄存器或IY寄存器的那些程序。

如果所设置的BIOS是用来保存这些寄存器，那末，一个应用程序可以使用替换寄存器和IY寄存器。（见2.17节）。系统根据所提供的情况保留这些寄存器。SETUP用来提供或去掉这种方便性。（见4.10节）。运行在Intel8080或8085上的已经设计好的一些程序将不管这种方便性而进行运行。

· 使用重启指令和/或定位的那些程序。

CP/M应用程序不倾向于呆板地使用重启指令。这是因为它们一般都为中断而使用。在CPC464中，应用程序仅仅只使用RST6。RSTs1……5则是由固件所使用，如果采取专门的步骤的话，它们可以用于其他一些目的中（见2.16节）。RST7则是专为一些中断而保留的。

· 要求一个比39.5k还要大的TPA的那些程序。

在它们可能在CPC464中运行之前，将需要作一些修改。

送到屏幕的一些控制码的功能在CPC464的固件手册的附录VII中专作介绍。

键盘可以根据要求而重新进行配置。SETUP可以被用来定义所要求的键盘布置，即键应该送回什么。信息是被存储在

磁盘上的，并且在冷引导期间，对键盘进行初始化。系统根据提供的内容不重新配制键盘，键盘的安排情况在固件手册的附录I……IV中给出。

1. 2 AMSDOS

AMSDOS 即为 AMSTRAD 磁盘操作系统的缩写。AMSDOS 已经被设计成能使 BASIC 程序与使用磁带文件的完全一样的方式去使用磁盘文件。不管什么时候，只要 DDI-1接口一旦联上，所有的BASIC文件操作都更改成磁盘而不是磁带了。AMSDOS使用了与CP/M一样的文件结构，因此，文件可以在这两个系统之间随便进行互换。

下面这些BASIC命令都将与磁盘一起工作：

LOAD
RUN
SAVE
CHAIN
MERGE
CHAIN MERGE
OPENIN
OPENOUT
CLOSEIN
CLOSEOUT
CAT
EOF
INPUT #9
LINE INPUT #9

WRITE #9

LIST #9.

可以看出，最主要一点是文件名必须与CP/M的会话一致才行。

磁带系统仍可以使用下面一些命令：

| TAPE——所有的文件操作转换回磁带。

| DISC——转换回到磁盘。

| TAPE-IN——输入文件操作转换回到磁带。

| DISC-IN——输入文件操作转换到磁盘。

| TAPE-OUT 输出操作转换到磁带。

| DISC-OUT 输出操作转换到磁盘。

CAT命令用于显示磁盘的目录。它是以字母的次序进行存储，显示出每一个文件的尺寸，在磁盘上所留下的空间等等。

利用外部命令| REN和| ERA 可以改变文件的名称和删除文件。例如，删除一个磁盘文件TEXT-DOC

FILENAME$ = "TEXT-DOC"

| ERA, @FILENAME$

要把文件名从RED-BAS改为GREEN-BAS，则：

OLDNAME$ "RED_BAS"

NEWNAME$ = "GREEN_BAS"

| REN, @NEWNAM$, @OLDNAME$

由于字符串必须通过地址送给外部命令，所以使用了有点回旋的语句。

如果表示驱动器的字母作为一个文件名发出，则驱动A：便假设为是当前所约定的驱动器。对于两个驱动器的系统而言，|B将选择驱动B：作为约定的驱动器，而|A将再次选择

驱动A：。

如果有要求的话，利用↓DIR命令可以显示出部分目录。可以取一个含有通配符的文件名的可选参数。只有与这些文件名相对应的文件才被显示出来。这与CP/M中的 DIR命令 非常相似。例如，要显示一些_BAS作为结尾的文件：

FILENAME $ = "*_BAS"

| DIR，@FILENAM $

用户号是一种逻辑含义，它将磁盘分成 16 个 不 同的 "用户"，其编号为 0 ……15，每一个都有它本身的目录。这是一个CP/M的特点，也由ASMDOS提供。初始化后所约定的用户是 0 。它可以通过| USER 命令来改变。例如命 令 | USER，10将使初始化后的约定的用户变成为10。当随便规定一个文件名时，用户号可以予先放在名称的驱动部分位置上。例如，把一个文件从用户号 5 改名为用户号11：

OLDNAME $ = "5：OLDNAME"

NEWNAME $ = "11：NEWNAME"

| REN，@NEWNAME，@OLDNAME。

USER命令在CP/M中有一对，但是，除了在AMSDOS 下之外，一个用户不可能规定为是一个文件名的一部分。不论什么时候，用户号可以与所使用的驱动名称合在一起。例如，要显示驱动器 B ：中用户号为 7 中的全部文件的目录，这些文件以字母作为开头，并且都有_HEX。

FILENAME $ = "7B：Z*_HEX"

| DIR， aFILENAME $

AMSDOS通过截夺大多数磁带固件入口而工作，因此，通过 BASIC 所装入的一个机器码程序也可以使用 AMSDOS。AMSDOS为了其本身的目的需要一个RAM 区，当AMSDOS正在使用一个机器码程序时，必须当心不要使该程序写到这个RAM区中去。有关细节问题可考阅3．9节。

AMSDOS 使用CP/MBIOS 对磁盘进行存取，某些扩充的BIOS程序也用在AMSDOS中，可参阅3．8和2．14节。

1．3 实用程序

磁盘系统提供了大量的实用程序，它们都运行在CP/M2.2下。这些程序的有些部分是由数据研究公司所提供，而且是一些标准的CP/M实用程序,另外，一些程序是由AMSTRAD专门为CPC464所开发的，因此不应该在其他CP/M系统中使用。还提供有程序化语言DR LOGO，见 AMSTRAD DDI-1 用户指令手册和“LOGO指南”SOFT160。

AMSDOS_COM（AMSTRAD）

从CP/M返回到AMSDOS和BASIC。与此相反的操作则是 I CPM。

ASM_COM（数据研究公司的命令）

8080的汇编程序，虽然在 CPC464 机器中所使用的处理器是Z80，但由于Z80支持了8080的全部操作码，所以仍可以使用8080的汇编程序。与此相反的操作没有配置。

BOOTGEN_COM（AMSTRAD）

将引导程序和扇区配置从一个磁盘复制到另一个磁盘。

CLOAD_COM（AMSTRAD）

将一个文件从磁带复制到磁盘。

CSAVE_COM（AMSTRAD）

将一个文件从磁盘复制到磁带。

CHKDISC_COM（AMSTRAD）

用两个驱动器对两个磁盘的内容进行比较。

COPYDISC_COM（AMSTRAD）

用两个驱动器将一个磁盘的内容复制到另一个磁盘上。

DDT_COM （DR 公司 的 命令， 为了 使 用 RST6，由
ISTRAD进行了修改）。

动态调试工具，8080的调试程序。

DISCCHK_COM（AMSTRAD）

用一个驱动器对两个磁盘进行比较。

DISCCOPY_COM（AMSTRAD）

用一个驱动器将一个磁盘复制到另一个磁盘。

DUMP_COM（DR公司）

以十六进制格式在屏幕上显示一个文件。

ED_COM（DR公司）

文本编辑程序。

FILECOPY_COM（AMSTRAD）
用一个驱动器将一些文件从一个磁盘复制到另一个磁盘。

FORMAT_COM（AMSTRAD）
对一个磁盘进行格式化。

LOAD_COM（DR公司）
以Intel 十六进制格式读一个文件，并产生一个.COM 文件。

MOVECPM_COM（DR公司的命令，由于页面的边界问题，经AMSTRAD进行了修改）。
构成一个任意大小的CP/M系统。用以形成RSX 的空间。

PIP_COM（DR公司）
外围设备的交换程序，在磁盘和其他外围设备之间复制一些文件。

SETUP_COM（AMSTRAD）
在扇区中改变一些参数。

STAT_COM（DR公司）
在文件，磁盘，用户和IOBYTE中给出一些命令，它也可以改变IOBYTE。

SUBMIT_COM（DR公司）

从一个文件上而不是从键盘上取CP/M的命令。

SYSGEN_COM（AMSTRAD）

把CP/M系统写到系统磁道上。

XSUB_COM（DR公司）

与SUBMIT_COM一起使用，对各个程序提供输入缓冲。

第二章 CP/M2.2和BIOS

本章对CPC464中所提供的CP/M2.2进行介绍，特别说明如何来使用固件跳转块和扩充的 BIOS 跳转块中的许多可以应用的方便性。读者应该对CP/M和它的结构有一定的知识。(见"CP/M指南" SOFT159)。

2.1CP/M的结构

CP/M含有三个软件模块：控制命令处理程序(CCP)，基本的磁盘操作系统(BDOS)以及基本的输入输出系统(BIOS)。(注意：这里所指的"基本的"与BASIC语言毫无关系)。CCP和BDOS是由DR公司提供的与机器本身无关的一部分CP/M。BIOS则是与机器无关的而涉及到所有低级设备驱动，特别是磁盘驱动方面的东西。BIOS驻留在DD1-1接口上的扩充ROM中。CCP和BDOS则驻留在系统格式盘的头上两道磁道上，并且当热引导执行时，由BIOS将它们装入到RAM中。

BIOS ROM的起始地址是#C000，结束地址是在#FFFF，也即是说，它复盖了屏幕RAM。ROM也存放AMSDOS以及部分DRLOGO。

从地址#AD33到#BFFF的RAM一般都保留不用。它存放了固件跳转块，固件以及BIOS的数据区。

剩下的存储区由CP/M和驻留系统扩充(RSX)所使用。在系统中，对任何RSX都不提供空间，但是，可以用MOVCPM

COM这条命令来建立起一个具有 RSX 所需要的空间的一个比较小的CP/M系统

BDOS通过BIOS 跳转块与 BIOS 进行联通。这在BDOS后面紧接着指明的。

从地址＃0040到＃004F这16个字节是为DRLOGO所 保 留的，BIOS不使用这个空间。

2。2　存储器安排图

在存储器安排中，CP/M尽可能地放在较高的地址区 中。

地址	存储区	说明
＃FFFF	BIOS　ROM	＃C000……＃FFFF；高位ROM/屏幕RAM。
＃COOO	BIOS　堆栈	
＃BEOO	BIOS　扩充跳转块	
＃BE80	固件和 BIOS 的变量	
＃AD33	BIOS　跳转块	RSX在这里进入，CCP和BDOS间下移
＃AD00	BDOS	
＃9F06	CCP	
＃9700	TBA	CCP可以被TPA复盖。
＃0100	页 0	＃0000……＃3FFF；低位ROM.
＃0000		

2。3冷引导

术语“冷引导”表示机器已加电或复位之后首次调用CP/M。

可以用下面两种方式中的一种来调用CP/M：当机器加电时自动调用；或是发出一个外部命令| CPM。

使用的方法取决于磁盘接口块上的不同跳接，BIOS的地址或置成0，或置成7。如果ROM地址设置为0，则当机器加电后将自动地进行BIOS冷引导。如果ROM地址设置成7，则需要发出一个外部命令| CPM才能调用CP/M。DDI-1是在ROM地址设置为7时才提供的。关于如何来改变ROM地址的详细介绍，可参阅5。1节。

BIOS冷引导一旦进行了调用，固件便通过调用MC START PROGRAM而复位，这样，位于后台ROM中的和RSX中的任何RAM都将丢失，并且所有的节拍链亦都复位。

BIOS冷引导程序第一次把BIOS初始化成一个“最小的”工作状态，从磁盘上的一个引导扇区把一个引导程序送到存储器，其起始地址为#0100。如果在引导扇区中，所有的字节相同，那末便会显示出一个出错信息。否则的话，便认为扇区中含有一个有效的引导程序。

“最小的”工作状态指引导程序能足以运行CP/M，但不执行下列的任何一件事：显示一个信息，初始化串行接口，设置任何键盘的转换等。

标准的引导程序通过磁盘装入扇区并完成进一步的初始化工作。这就执行了“两级”初始化，因此，如果需要的话，引导程序可以引出许多不同的操作系统。（见2。12节）。

一旦BIOS被初始化，便完成一个热引导以用来装入CCP和BDOS。

2.4 热引导

术语“热引导”表示了在一个暂存的程序已经运之后，或是用户为了改变一个磁盘而发出了Control-C命令之后，重新装入CP/M。

热引导可以在初始化之后立即调用，也可以在跳到地址0，或调用BDOS 系统复位功能之后立即调用。热引导的最主要的功能是把BDOS和CCP 装到存储器地址是通过跳转指令在CCP开始处进行计算得到的。这是通过将把JMP的目的地址减去#035C而给出CCP 的起始地址的。由于 CCP 的位置在每次热引导时都重新计算得到，所以不同大小的CP/M 可以相互进行热引导。

如果在CCP的第一个扇区中都具有一样的数值，则便显示出一个出错信息。否则，便认为系统磁道中含有一个有效的CCP和BDOS。

热引导程序一开始跳转到#0000和#0005，并且把 BIOS跳转块复制到RAM中去，这是在BDOS之后马上执行的。（见2.2节）。

2.5 驻留系统扩充（RSX）

RSX 允许用户保持一些如特殊字符输入/输出等程序在存储器中。它们必须放在BIOS变量区以下的存储区中（BIOS 变量区的地址开始于#AD33），但是还必须在BIOS的跳转块的RAM复制区的结束地址的上面的区域。在CP/M系统中，不对

RSX提供空间。如果需要RSX，则用户必须减小TPA的尺寸来满足所有的RSX所要求的空间量。注意，MOVCPM_COM命令只允许用户把CP/M重新安排在256个字节（页）的边界上。

设Srsx表示RSX以页而要求的尺寸(一页等于256个字节)，则RSX区将起始于＃AD33-（256*Srsx）而结束在＃AD32。

BIOS不对任何后面ROM进行初始化，AMSDOS不能在CP/M下开始工作。

在固件ROM 的版本1中，有如下一些限制：

任何一个所要求的的后台 ROM 必须通过调用KL INIT BAK而进行初始化。使用KL ROM WALK 会使磁盘扩充ROM重新被初始化，这样将会引起系统发生闭锁。ROM的动态变量的存储区必须在RSX区中，即介于 BIOS 跳转块尾部与BIOS的变量区以下之间。

如果ROM地址设置为0，且不调用 BASIC，那末就将不采用AMSDOS。

2.6 MOVCPM_COM的使用

为了将一个RSX插入到存储器里,就需要将CP/M系统向下移动。CP/M的MOVCPM_COM命令将根据CP/M 所要求的尺寸，在存储器中建立起一个映象，然后用命令 SYSGEN_COM将这个映象写入到磁盘上去。（见4.11节）。

MOVCPM命令要求规定好CP/M的尺寸。以前，这个尺寸表示了对含有1.5K的BIOS的原始的CP/M 所采用的全部RAM的大小。如令,为MOVCPM 所提供的存储字节几乎是一个不可思议的量了！MOVCPM要求所提供的尺寸以页（一页等于256个字节）来表示，其计算方法如下：

设Scpm表示MOVCPM命令中所要求的以页来表示的CP/M的尺寸。

设Srsx表示RSX所要求的页的尺寸。

因此，　Scpm = 179 - Srsx。

又设：Bstart表示BDOS的起始地址，

　　Bentry表示BDOS的入口点地址

因此，　Bstart = （Scpm - 20）*256

　　　　Bentry = Bstart + 6

TPA的尺寸则为Bentry - #0100。

为了运行MOVCPM，则送入：

MOVCPM nnn*。

这里nnn是所要求的尺寸，它必须在64…179范围之内。

星号*必须要加上，它与尺寸数之间必须有一个空格。

通过下面这样一条命令，便可建立CP/M系统：

MOVCPM 179*

2.7 IOBYTE

CP/M支持了四个逻辑的字符I/o设备，它们称为：

CONSOLE，READER，PUNCH　和LIST。

这四个逻辑设备分别映象到不同的物理设备上。在任何一个时候，映象到物理设备的当前的逻辑设备都在一个地方被定义，这个地方就称之为IOBYTE。暂存实用程序STAT可以用来改变IOBYTE。

CONSOLE可以被分配到：

TTY：（专用I/o设备0）

CRT：（键盘和屏幕）

BAT：（从READER输入并输出到LIST）

UC1：（专用I/o设备1）

READER可以分配到：

TTY：（专用I/o设备0）

PTR：（文件输入的结束）

UR1：（专用I/o设备1）

UR2：（键盘）

PUNCH可以分配到：

TTY：（专用I/o设备0）

PTP：（零输出）

UP1：（专用I/o设备1）

UP2：（屏幕）

LIST可以分配到：

TTY：（专用I/o设备0）

CRT：（屏幕）

LPT：（centronics 口）

UL1：（专用I/o设备1）

键盘输入设备利用固件程序KM WAIT CHAR从键盘上读取字符。

屏幕输出设备利用固件程序TXT OUTPUT来显示文本以及对控制字符进行处理。关于控制字符的详细介绍可参阅固件的有关资料）。

文件结束输入设备一般送回文件的结束字符#1A。零输出设备将送到它的所有字符全部给予废弃。

两个专用I/o设备都用来驱动Z80的SIO的A通道和B通道。每一个设备都有四个驱动程序，它们都是通过扩充的跳转块（见2．14节）而间接工作。为了能使用一个 RSX 设备驱动程序，用户可以对这些跳转块进行修改。

2．8 BIOS用户间的相互影响

一般来讲，如果 BIOS 不对磁盘进行存取的话，用户可以在键盘上随意地打入命令。而在对磁盘存取时，则必须将中断禁止并且有可能会丢失掉一些字符。

有些字符是专门由 BIOS 来处理的（见2．14节）。当用Centronics或串行接口来执行字符的输入/输出时，BIOS 用以下控制字符来对键盘进行查询：

CONTROL-C

为了把系统从闭锁状态前停下来，在下列条件之下，用户可以按Control-C来执行一个热引导：

a） 当正在等待对串行接口输出一个字符时。

b） 当正在等待对串行接口输入一个字符时。

c） 当正在等待对Centronics口送一个字符时。

CONTROL-S

当对串行接口发送或接受一些字符的时候，要想中止CP/M用户可以按CONTROL-S。一旦中止住后，用户必须按下任何一个键来继续进行BIOS的操作。Control-S以及所按的字符都丢失。

CONTROL-Z

当正在等待为串行接口输入一个字符时，用户可以按CONTROL-Z，这将引起送回一个CONTROL-Z以对程序进行调用。

2．9 初始命令缓冲器

当调用CP/M时，键盘输入设备从一个专用的称之为初始命令缓冲器中读一些字符。SETUP程序允许用户对初始命令冲器进行初始化，指定许多CP/M在冷引导时能自动执行的命令。

BIOS程序CONST不反映出初始命令缓冲器的一些状态。这样，对control-c查询的一些程序将不挂空初始命令缓冲器。尤其是在2．8节中所述的BIOS用户相互间的作用，不去从初始命令缓冲器中取一些字符。

2．10 BIOS的一些信息

所有的BIOS信息都通过TXT OUTPUT而送到屏幕上。它们不通过IOBYTE。

BIOS的一些信息都是跟有问号的Retry，Ignore or Cancel？。

系统然后把一些字符丢掉，开启光标并等待用户送入R，I或C，除此之外，任何其他事情都将引起发出“嘟嘟”声。

送入R将使BIOS重复地进行操作。

送入I将使BIOS继续执行下去，当然如果不发生问题的话。

送入C将使BIOS中止掉操作。这种情况通常会引起一个BIOS

的出错信息。

用户送入R、I或C之后，光标便关闭。

AMSTRAD BIOS有如下一些信息：

Drive 〈n〉：disc missing

这里〈n〉是驱动器的标志符，它或者为A，或者为B。当BIOS企图对一个还未插入磁盘或对一个不存在的驱动器进行存取时，便发出这个信息。

Failed to load boot sector

在冷引导期间，当引导扇面读得不对时，或在引导扇区中的所有字节都具有相同的数值时便出现这个信息。

Failed to Load CP/M

在热引导期间，当CCP或BDOS的扇区读得不对时，或在第一个CCP扇区中所有的字节都具有相同的数值时便出现这个信息。

Drive〈n〉：disc is write protected

这里〈n〉是驱动器标志号，它或为A，或为B。当BIOS企图对一个有写保护的磁盘进行写入时，便出现这个信息。如果用户想在这个磁盘上写入数据，那末用户应该把这个磁盘拿出来，去掉写保护片，再重新将它插进驱动器，然后再按R才行。

Drive〈n〉：read fail

这里〈n〉是驱动器标志号，它或为A，或为B。当对磁盘

进行读出时，如果出现了硬件上的错误，便发生该信息。如果对一个具有错误格式的磁盘进行读出，例如：想对一个只有数据格式的磁盘进行热引导，则也会产生这一个出错信息。

Drive<n>：write fail

这里<n>是驱动器标志号，它或为A，或为B。当对磁盘写入时，发生了硬件上的错误后就产生出这一个信息。

在读或写时发生故障时，建议用户将磁盘从驱动器取出，然后再插到驱动器中，再按“R”。万一磁盘位置插得不好，那末这样做可能是有帮助的。如果问题还不能得到解决，那末只好用其他方法来想法取得数据。为此，建议用户一定要使用磁盘的备用复制盘。

2．11 磁盘的格式

BIOS支持了三种不同的磁盘格式：

SYSTEM 格式
DATA ONLY格式和
IBM格式。

BIOS自动地对一个磁盘的格式进行检测。在CP/M下对驱动器A在，一个热引导情况下进行，而对驱动器来讲，是在它第一次进行存取时进行。在AMSDOS下，是在每一次磁盘存取不打开的文件时进行的。为了允许进行自动检测，每一种格式都具有唯一的扇面数。

3英寸的磁盘都是双面的，但根据用户插入磁盘的方法，一次只能存取一个面。在两个面上可能有不相同的格式。

对所有的格式都共同的：
单面（一个3英寸磁盘的两个面分别进行处理的）。
物理扇区的尺寸为512个字节。
40个磁道，其编号从0到39。
CP/M块的尺寸为1024个字节。
64个目录。

系统格式：
每一个磁道有9个扇区，其编号为#41到#49。
2个保留磁道。

系统格式是所支持的主要一种格式，因而对一个系统格式的磁盘只能送入CP/M（冷引导和热引导）。保留磁道用法如下：

磁道0，扇区#41：冷扇区。
磁道0，扇区#42：格式扇区。
磁道0，扇区#43…#47：不用。
磁道0，扇区#48…#49，以及磁道1，扇区#41…#49：用于CCP和BDOS

注意："买主"格式是系统格式中的一种特殊版本，它在两个保留磁道上不含有任何的软件。它倾向于使用于对软件进行分配。

纯数据格式：
每一个磁道有9个扇区，编号为#C1…#C9。
0保留磁道。

这种格式主要用于进一步增强用，对于使用CP/M来讲不推荐使用，因为对它不能执行热引导。如果仅仅只使用 AMSDOS，那末，多少还可以利用一些磁盘上的空间。

IBM 格式：

每个磁道有8个扇区，编号为1…8。

一个保留磁道。

没有插入的扇区。

这种格式在逻辑上是与在IBMPC机上CP/M所使用的单面格式相同的。它主要为专门的目的而使用，因为对它进行热引导是不可能的，所以也不能推荐使用。

2.12 引导扇区

为了今后能够提供一些非CP/M的系统，在装入 CP/M 之前，通过一个对磁盘进行读出的引导程序来执行 BIOS的初始化工作。在非CP/M 的情况下，引导程序不跳到热引导程序中去，而是进入到它自己本身的通路上去，这是利用 BIOS 以及所要求的固件程序来实现的。

引导扇区是0磁道上的第 #41号扇区。

在冷引导期间，BIOS被初始化成最小的状态，即：

所有在ROM中的程序都是BIOS跳转块的拷贝，并且所有在扩充块中的程序都被采用。

替换寄存器和IY寄存器都被省掉。

中断通过BIOS而间接使用，并在BIOS的堆栈中运行。

磁盘信息被使能。

初始命令缓冲器为空。

在＃0003处的IOBYTE被初始化成＃81（LST：＝LPT：，PUN：＝TTY：，RDR：＝TTY：，CON：＝CRT）。

在＃0004处的当前驱动器被初始化成＃00。

串行接口不进行初始化。

CCP和BDOS都不在存储器中。

BIOS RAM跳转块不在存储器中。

在＃0000及0005的CP/M跳转程序都不进行初始化。

引导扇区被读出并装入到存储器地址＃0100中，堆栈指针被初始化成＃AD33，在BIOS的数据区下面的一个数值以及引导程序被送入到＃0100。引导程序可以使用从＃0100到＃AD32这一段存储区。

为了运行CP/M，引导程序必须跳转到在ROM跳转块中的热引导入口上去。

引导程序具有如下这些条件：

入口：

SP＝最高的可利用的地址＋1（对堆栈来讲为最低的地方）。

BC＝复制BIOS跳转块的ROM地址。（BOOT）

出口：

程序应该跳到上述跳转块中的WBOOT入口。

复制BIOS跳转块的ROM不应该在其他时候使用（事实上也只有引导程序才知道它在什么地方）。

2.13 格式扇区

所提供的引导程序根据磁盘送入格式扇区。它含有各种各样的系统参数，这些系统参数都是用来对BIOS进行初始化而

用的，这些参数还可以通过 SETUP_COM 命令来进行 修改。（见4．10节）。

在O磁道＃42扇区上所含有就是格式扇区。

格式：

字节0 = ＃35，字节1 = ＃12为磁盘特征号。

这个特征号允许引导程序检查格式扇区有没有“建立”过。

字节2……3为驱动器马达的开启延迟时间。

这一段时间为驱动器开始运转到开始进行读写操作的等待时间。它以1/50秒为单位，如果这个时间为零，则表示系统将闭锁。

字节4……5为驱动器马达的关闭延迟时间。

这一段时间为读写操作之后和驱动器马达关闭之前的等待时间。它也以1/50秒为单位。如果这个时间为零，则表示马达一直保持运转。

字节6为步进速率。

它即为磁盘移动的步进速率。它必须为2…32毫秒范围内的偶数值。

字节7为IOBYTE。

它是CP/MIOBYTE的初始值。

字节8为BIOS磁盘信息的使能/禁止。

＃00⇨使能BIOS磁盘信息。

＃FF⇨关闭BIOS磁盘信息。

字节9为替换寄存器和IY寄存器的使能/禁止。

＃00⇨使能这些寄存器（慢方式）。

＃FF⇨禁止这些寄存器（快方式）。

字节10…21都用于对串行接口的初始化（如果有串行接口的话）。

字节10：SIO通道A写寄存器4（方式寄存器）。

字节11：SIO通道A写寄存器5（Tx寄存器）。

字节12：SIO通道A写寄存器3（Rx寄存器）。

字节13：SIO通道B写寄存器4（方式寄存器）。

字节14：SIO通道B写寄存器6（Tx寄存器）。

字节15：SIO通道B写寄存器3（Rx寄存器）。

字节16…17：8253定时器0（SIO通道A的Tx波特率）。

字节18…19：8253定时器1（SIO通道A的Rx波特率）。

字节20…21：8253定时器2（SIO通道B的Tx和Rx 波特率）。

字节22为命令保存缓冲器。

#00⇨　当一个键被按下时，初始命令缓冲器将被刷新。

#FF⇨　当一个键被按下时，初始命令缓冲器将被刷新。

字节23…99作为保留用。

这些字节都不使用，并且全部置成#00。

下列这些内容都具有可变长度，并从字节100开始放起。

起始字符串。

是输出到屏幕的一串字符，该字符串以一个 $符号来表示结束。

打印机加电字符串。

该字符串被送到CP/M的列表设备，那它是间接地通过IOBYTE 进行的。与字节的设置有关。

字节0为字符串长度。0…255。

字节1开始为字符串。

"正常的"键盘译码。

该表重新定义了由不按下"换档键"和不按下"控制键"时所产生的代码。在此表中，每一个入口均为两个字节长。第一个字节为键码，第二个字节是产生这个代码的键号。有关键号的说明可参阅附录Ⅰ。

表的格式：

字节0，入口号，0…255。

字节1开始为各个入口。

入口的格式：

字节0为键码。

字节1为键号。

键盘"换档"时的译码。

当按下"换档"键时，该表对所产生的代码进行重新定义。表中的每一个入口都为两个字节长。其第一个字节是键码，第二个字节是产生这个代码的键号。关于键号的说明可参阅附录Ⅰ。

表的格式：

字节0：入口号，0…255。

字节1开始起为各个入口。

入口格式：

字节0：键码。

字节1：键号。

键盘处于"控制"情况下的译码。

当按下"控制"键时，该表对所产生的代码进行重新定义。表中每一个入口都为两个字节长度。第一个字节是键码，

第二个字节是将产生该代码的键号。关于键号的说明可参阅附录Ⅰ。

表的格式：

字节0：入口号，0…255。

字节1开始起为各个入口。

入口格式：

字节0为键码。

字节1为键号。

键盘的扩充。

该表定义了键盘的扩充字符串（见3.7节）。

表的格式：

字节0：入口号，0…255。

字节1开始起为各个入口。

入口格式：

字节0：扩充标志。

字节1：扩充字符串的长度。

字节2开始起为扩充字符串。

剩下来的格式扇区置成#00。

2.14 BIOS跳转块

AMSTRAD BIOS有两种跳转块，一种是标准的CP/M跳转块，一种是扩展的跳转块。扩展的跳转块起始于#BE80并包含了一些如物理扇区的读写这样的附加特性的跳转程序。当CP/M为热引导时，标准的BIOS跳转块立即拷贝到RAM中，并位于BDOS的后面。在RAM跳转块中，热引导入口地址以字的形式存放在#0001。

标准BIOS跳转块的入口条件及出口条件都是如下所列出的，在CP/M改变了规范而所定义的那些东西的再设置。在标准的BIOS跳转块中的所有的程序都有选择地根据SET REG SAVE程序保存了Z80的替换寄存器和IY寄存器。

读

写

这两个程序在读或写失败时，于累加器中送回一些硬件的出错状态。这些状态位的含义如下：

位7：磁道的结束。　　—这一位一般都应该设置。

位6：不使用。　　—通常为1。

位5：数据错。　　—CRC错（数据或ID段）。

位4：越时错。

位3：驱动器未准备好。　　—在驱动器中没有磁盘。

位2：无数据。　　—找不到扇区。

位1：不能写入。　　—驱动器为写保护。

位0：地址标记丢失。

除上述状态字节之外，HL寄存器对存放了在计算μPD 765A磁盘控制器的相位期间时所接收到的字节的一个缓冲器的地址。第一个字节是下面所有的字节的计数值。（见制造厂的资料）。

磁盘的选择。

如果E的位0等于零，则表示E在第一次对磁盘进行存取，此时，BIOS将通过从当前的磁道中读出一个ID头标来决定磁盘的格式。完成此项工作后，相应的扩充的DPB被进行修改，以满足所检测到的格式。

扩充跳转块的起始地址为#BE80，并且有如下一些种类：

#BE80 JMP SET MESSAGE ;
使能/禁止磁盘出错信息。
#BE83 JMP SETUP DISC ;
对驱动器的参数进行初始化。
#BE86 JMP SELECT FORMAT ;
选择一个标准格式。
#BE89 JMP READ SECTOR ;
从磁盘上读一个扇区。
#BE8C JMP WRITE SECTOR;
对磁盘写一个扇区。
#BE8F JMP FORMAT TRACK ;
对一个完整的磁道进行格式化。
#BE92 JMP MOVE TRACK ;
移到所指定的磁道。
#BE95 JMP GET DR STATUS ;
给出驱动器的状态（ μpD765A的寄存器3）。
#BE98 JMP SET RETRY COUNT ;
设置一个出错再检查的数。
#BE9B JMP ENTER FIRMWARE ;
调用一个固件程序。
#BE9E JMP SET REG SAVE ;
建立/清除替换寄存器
#BEA1 JMP SET SIO ;
对串行接口进行初始化。
#BEA4 JMP SET CMND BUFFER ;
对命令缓冲器进行初始化。

下面8个程序都是专用字符输入/输出设备的间接跳转程序。BISO使用这些跳转程序来对这些设备进行访问。如果需要的话，用户可以将这些跳转程序修改成一个RSX的驱动程序它们全部都约定为串行接口驱动程序。设备0是串行接口的通道A，设备1则是通道B。

#BEA7 JMP DO IN STATUS ；
驱动器0的输入状态。

#BEAA JMP DO IN ；
从驱动器0输入。

#BEAD JMP DO OUT STATUS ；
驱动器0的输出状态。

#BEB0 JMP DO OUT ；
输出到驱动器0。

#BEB3 JMP D1 IN STATUS ；
驱动器1的输入状态。

#BEB6 JMP D1 IN ；
从驱动器1输入。

#BEB9 JMP D1 OUT STATUS ；
驱动器1的输出状态。

#BEBC JMP D1 OUT ；
输出到驱动器1。

在扩充跳转块中的每一个入口所完成的功能与它们的入口条件和出口条件一起将在下面介绍。

在返回时，这些程序将调用一些固件程序，因此，它们必须能象固件程序那样自行进行处理。（见2。17节）。

0 SET MESSAGE #BE80

使能/禁止 磁盘的出错信息。

功能：当磁盘出错信息被使能，且出现了一个错误时，BIOS将在屏幕上显示出一些出错信息。当禁止时，则不显示出错信息。

入口条件：A＝#00⇨使能磁盘的出错信息。

A＝#FF⇨禁止磁盘的出错信息。

出口条件：A＝原先的状态。

所有标志位及HL不定。

其余各寄存器保留不变。

注：加电时，其状态为“使能”。

有关入口：

SET RETRY COUNT。

1 SETUP DISC #BE83

复位各种磁盘的参数。

功能：为马达启动，马达关闭，关闭写电流以及磁头的定位时间等设置一些数值。对软盘控制器送一个 SPECIFY 命令。

入口条件：HL＝参数块的地址。

参数块的格式如下：

字节0，1： 马达开启时间，以20毫秒为单位。

字节2，3： 马达关闭时间，以20毫秒为单位。

字节4： 写电流关闭时间，以10微秒为单位。

字节5： 磁头定位时间，以1毫秒为单位。

字节6： 步进速率，以1毫秒为单位。

字节7：　磁头卸载延迟（作为每一个upD765A的SPECIFY命令）。

字节8：　位7……1为磁头装载延迟，位0为非DMA方式（作为每一个upD765A的SPECIFY命令）。

出口条件：AF，BC，DE及HL都为不定。
其余各寄存器保留。

注：所给出的值都是对两个驱动器而言的。当使用两个不同的驱动器时，用最慢的两个时间。

在加电之后，上述数值为：

马达启动时间为50　（1秒）
马达关闭时间为250　（5秒）
写电流关闭时间而175　（1.75毫秒）
磁头定位时间为15　（15毫秒）
步进速率为12　（12毫秒）
磁头装载时间为1
磁头卸载时间为1
非DMA方式为1。

马达启动时间若为零，则系统便闭锁。而马达关闭时间若为零，则马达一直不关闭。

标准的引导程序调用这一个程序来复位一些磁盘的参数，这些参数都是在格式扇区中所规定要有的，如马达开启时间，马达关闭时间以及步进速率等。

2　SELECT FORMAT　　　　　　#BE86

选择一种AMSTRAD的格式，而不考虑磁盘实际的格式。

功能：这一个程序对所给定格式的扩充的磁盘参数进行初始化。一般来讲，当调用SELDSK时，BIOS会自动地利用寻找扇区号的方法对一个磁盘的格式进行检测。但是，对这样一个格式化的程序来讲，则必须予先设置好格式。

入口条件：A＝所要求的格式的数的第一个段。

#41⇨系统格式。

#C1⇨纯数据格式。

#01⇨IBM格式。

E＝驱动器。

#00⇨A。

#01⇨B。

出口条件：AF、BC、DE及HL均不定。

其余各寄存器保留。

注：扩充的磁盘参数块中的字节0…21全部进行复位，所有原先所设置的值全被丢失。字节22…24（磁道、排列标志、自动选择标志）都不受影响。（见2。15节）

为了设置一个非AMSTRAD格式，用户可以对扩充的磁盘参数块进行直接修改。

3　READ SECTOR　　　　　　#BE89

从磁盘上读一个物理扇区。

功能：

把指定的扇区的内容读到存储器。

入口条件：

HL＝扇区缓冲器的地址。

E＝驱动器号

#OO⇨A：

#O1⇨B：

D＝磁道号

C＝扇区号

出口条件：如果进位位为真，则扇区的读出正确。

且A＝0，

HL保留。

如果进位位为假，则扇区读出不正确。

且A＝前面所定义的出错状态码。

HL＝出错状态缓冲器的地址。

通常，其它标志位均不定，其余各寄存器保留。

有关入口：

WRITE SECTOR。

4 WRITE SECTOR #BE8C

对磁盘的一个物理扇区进行写入。

功能：

从存储器中将数值写入到所指定的扇区。

入口条件：

HL＝扇区缓冲器的地址。

E＝驱动器号。

#00⇨A：

#01⇨B：

D＝磁道号。

C＝扇区号。

出口条件：如果进位位为真，则扇区的写入正确。

且A＝0，HL保留。

如果进位位为假，则写入不正确。

且A＝前面所述的出错状态字节。

HL＝出错状态缓冲器的地址。

通常：其他标志位不定，其余各寄存器保留。

注：

扇区缓冲器可以在ROM下面。

有关入口：

READ SECTOR。

5 FORMAT4TRACK #BE8F

对一个完整的磁道进行格式化。

功能：

对一个磁道进行格式化。

入口条件：HL＝头标信息缓冲器地址。

E＝驱动器号。

#00⇨A：

#01⇨B：

D＝磁道号。

头标信息的格式：

第一扇区的扇区入口。

第二扇区的扇区入口。

⋮

最后一个扇区的扇区入口。

扇区入口的格式：

字节0：磁道号。

字节1：头号。

字节2：扇区号

字节3：$\log_2$（扇区大小）－7。

出口条件：如果进位位为真，则：

磁道格式化完成。

A＝0

HL保留

如果进位位为假，则：

磁道格式化不正确，

A＝前面所述的出错状态字节。

HL＝出错状态缓冲器的地址。

通常，其他标志位不定，其余各寄存器保留。

注：扩充的DPB必须予置成所要求的格式。

（见SELECT FORMAT）。

有关入口：

SELECT FORMAT。

6 MOVE TRACK #BE92

不加检验地移到指定的磁道。

功能：

将磁头移到指定的磁道上。

入口条件：

E = 驱动器号

#00⇨A：

#01⇨B：

D = 磁道号

出口条件：如果进位位为真，则：

移动得正确。

A = 0。

HL保留。

若进位位为假，则：

移动得不正确。

A = 前面所定义的出错状态字节。

HL = 出错状态缓冲器的地址。

通常： 其他标志位不定，其余各寄存器保留。

注： 这一个程序主要用来作为一种诊断手段，但由于读/写/格式化这些程序都能正确地对相应的磁道进行寻找，所以一般情况下不必使用这一个程序。

7 GET DR STATUS #BE95

对所指定的驱动器送回一些状态

功能：这一个程序对所指定的驱动器送回下面所定义的状态到软盘控制器的寄存器 3 中。

位 7： 不定义。

位 6： 写保护——写保护信号为真。

位 5： 驱动器准备好——准备信号为真。

位 4： 磁道零——磁道零信号为真。

位 3： 不定义。

位 2： 磁头地址——通常为零。

位 1： 单位选择 1 ——通常为零。

位 0： 单位选择 0 ——当前所选择的驱动器。

入口条件：A = 驱动器号

00⇨A：

01⇨B：

出口条件：

如果进位位为真，则：

A = 上面所定义的驱动器状态字节。

HL保留。

如果进位位为假，则：

HL = 出错状态缓冲器地址， 其第二个字节为前面所定义的驱动器状态字节。

A不定。

通常，其他标志位不定，其余各寄存器保留。

注：这个程序用进位位来表示出口条件，进位的状态没有其它别的含意。

8 SET RETRY COUNT #BE98

设置重新再进行读/写/格式化的次数。

功能：在出错之后，对再重新进行一种操作的次数进行设

置。

入口条件：A = 重新操作的次数的新的数值。

出口条件：A = 重新再进行操作的原先的次数。

所有标志位和HL都不定。

注：重新再进行操作的型式如下，每一个“尝试”计为1，一直重复到操作成功或尝试次数已达到规定的次数时为止。

尝试
尝试
移入一个磁道，然后再退回
尝试
移出一个磁道，然后再退回
尝试
移到里面的磁道，然后再退回
尝试
尝试
移入一个磁道，然后再退回
尝试
移出一个磁道，然后再退回
尝试
移到外面的磁道，然后再退回
重复
规定值是16，那上述的循环进行两次。

9 ENTER FIRMWARE

调用一个固件程序。

功能：调用一个固件程序，应用程序必须由所调用的固件

程序按照要求将所有的寄存器建立好，应用程序的地址以一行参数送出。

所给定的程序运行在BIOS的堆栈中。

下面的一些程序不能用ENTER FIRMWARE进行调用：

ENTER FIRMWARE

SET REG SAVE

其他标准的BIOS程序。

关于什么时候使用这个程序，以及为什么要使用这个程序的说明，可参阅2、17节。

入口条件:AF、BC、DE,HL及IX由所调用的程序决定。

返回地址指向要调用的程序的地址。

出口条件：AF、BC、DE、HL及IX由每一个所调用的程序所决定。

其他各寄存器保留。

有关入口：**SET REG SAVE**

举例：

为了从键盘读一些字符，使用KM WAIT KEY：用TXT OUTPUT在屏幕上对它们作出回响。

```
LOOP: CALL   ENTERFIRMWARE
      DW     KMWAITKEY      ；一行参数
```

```
CALL   ENTERFIRMWARE
DW     TXTOOTPUT      ；一行参数

JP     LOOP
```

10 SET REG SAVE #BE8E

设置/清除替换寄存器和IY寄存器的保留。

功能：使能或禁止替换寄存器和IY寄存器的保留。（见2、17节和 ENTER FIRMWARE程序）

这个程序不能用 ENTER FIRMWARE程序调用。

入口条件：A＝#00⇨保留替换寄存器和IY寄存器。

A＝#FF⇨不保留替换寄存器和IY寄存器。

出口条件：A＝原先的值。

所有标志位及HL不定。

其余各寄存器保留

有关入口：**ENTER FIRWMARE**

11 SET SIO #BEA1

对串行接口进行复位和初始化。

功能：对串行接口SIO的两个通道进行复位和初始化。把8253初始化成为当作波特率发生器用。

入口条件：HL＝参数块的地址。

参数块的格式：

字节0：SIO 通道A写寄存器4（方式寄存器)。

字节1：SIO 通道A写寄存器5（t_x寄存器）。

字节2：SIO 通道A写寄存器3（r_x寄存器）。

字节3：SIO 通道B写寄存器4（方式寄存器）。

字节4：SIO 通道B写寄存器5（t_x寄存器）。

字节5：SIO 通道B写寄存器3（r_x寄存器）。

字节6、7：8253定时器0（SIO通道A的t_x波特率）。

字节8、9：8253定时器1（SIO通道A的r_x波特率）。

字节10、11：8253 定时器2（SIO通道 B的t_x和r_x波特率）。

出口条件：AF，BC，DE和HL不定。

其余各寄存器保留。

注：SIO和8253都不包含在DDI-1接口中。

参阅Zilog的SIO和8253的资料。

12 SET CWND BUFFER #BEA4

对初始命令缓冲器进行初始化。

功能：所提供的缓冲器被复制到 BIOS 的初始命令缓冲器中。当BIOS的程序 CONIN被调用时，它便从命令缓冲器中取出一些字符，一直取到缓冲器被取空时为止。

当从命令缓冲器中对命令进行读出时，任意按一个键，就会将命令缓冲器挂空。所显示的字符应该是所送回的下一个字符。所有依次的字符都将从键盘所取出。

入口条件：A = #00⇨当按下一个键时，命令缓冲器被丢

弃。

A = #FF⇨当按一个键，命令缓冲器仍保持。

HL = 缓冲器的地址。

缓冲器的格式：

字节0：字符0……128的数目。

字节1……128：字符本身。

出口条件：AF、BC、DE和HL不定。

其余各寄存器保留

注：虽然这个程序主要是在冷引导期间使用，但它还是可以在任意时候被调用，用来将缓冲器重新进行装填。

CONST 送回了键盘的状态，而不是送回初始命令缓冲的状态。这样一个对查询Control-C 的程序将不致挂空命令缓冲器。

13 DO IN STATUS #BEA7

测试一下专用字符I/O设备0，看看有没有可以利用的字符。

功能：测试一下，看看有没有一个可利用的字符。

入口条件：无条件。

出口条件：如果A = #FF，则有一个字符可用于输入。

如果A = #00，则没有可用于输入的字符。

通常，其余所有标志位及BC、DE、HL不定。

注：规定设备0是串行接口的通道A。

如果需要，这个跳转程序可以改为一个RSX的驱动程序。

有关入口：DO IN
DO OUT STATUS
DO OUT

14 DO IN

取一个字符

功能：从专用字符I/O设备0输入一个字符。如果没有一个字符可供使用，则一直等待到出现一个可使用的字符为止。

入口条件：无条件。

出口条件：A＝字符输入。

所有标志位及BC、DE及HL不定。

其余各寄存器保留。

注：规定设备0是串行接口的通道A。

如果需要，该跳转程序可以改成为一个RSX驱动程序。

有关入口：DO IN STATUS
DO OUT STATUS
DO OUT

15 DO OUT STATOS #BEAD

测试一下，看看输出是否准备好

功能：测试一下，看看专用字符I/O设备0是否准备输出一个字符。

入口条件：无条件。

出口条件：如果A＝＃FF，

则输出设备已准备好。

如果A＝＃00，

则输出设备处于忙。

通常，所有标志位及BC、DE、HL均不定。

其他各寄存器保留。

注：规定的设备0是串行接口的通道A。

该跳转程序在需要时可以改成为一个RSX的驱动程序。

有关入口：DO IN STATUS

DO IN

DO OUT

16　DO OUT

输出一个字符

功能：将一个字符输出到专用字符I/O设备0。如果设备为忙，则一直等待到它变为准备好为止。

入口条件：C＝要输出的字符。

出口条件：AF、BC、DE及HL不定。

其余各寄存器保留。

注：规定的设备0为串行接口的通道A。

该跳转程序如需要的话可以改为一个RSX的驱动程序。

有关入口：DO IN STATUS

DO IN

DO OUT STATUS

17　D1 IN STATUS　#BEB3

测试一下专用字符I/O设备1，看看有没有可使用字符

功能：测试一下，看看有没有可使用的字符。

入口条件：无条件。

出口条件：如果A＝#FF，则有一个字符可用于输入。

　　　　　如果A＝#00，则没有字符可用于输入。

通常，所有标志位及BC、DE、HL均不定。

注：规定的设备1是串行接口的通道B。

该跳转程序，如果需要，可改成一个RSX驱动程序。

有关入口：**D1 IN**

　　　　　D1 OUT STATUS

　　　　　D1 OUT

18　D1 IN　#BEB6

取一个字符。

功能：从专用字符I/O设备1输入一个字符。如果没有可使用的字符，则一直等待到出现一个为止。

入口条件：无条件。

出口条件：A＝输入字符。

　　　　　所有标志位及BC、DE、HL不定。

　　　　　其余各寄存器保留。

注：规定的设备1为串行接口的通道B。

该跳转程序，若需要时可以改成一个RSX驱动程序。

有关入口：D1 IN STATUS

D1 OUT STATUS

D1 OUT

19　　D1 OUT STATUS　　#BEB9

测试一下，看看输出是否准备好。

功能：测试一下，看看专用的字符I/O设备1是否准备输出一个字符。

入口条件：无条件

出口条件：若A = #FF，则输出设备已经准备好。

若A = #00，则输出设备处于忙。

通常，各标志位以及BC、DE、HL不定、其余各寄存器保留。

注：规定的设备1是串行接口通道B。

如果必要的话。该跳转程序可以改为一个RSX驱动程序。

有关入口：D1 IN STATUS

D1 IN

D1 OUT

20　　D1 OUT　　#BEBC

输出一个字符

功能：输出一个字符到专用的字符I/O设备。如果设备处于忙，则一直等到它变为准备好。

入口条件：C = 输出的字符。

出口条件：AF、BC DE及HL不定。

所有其他寄存器保留。

注：规定的设备1是串行接口通道B。

该跳转程序，若有必要时，可以改成为一个RSX驱动程序。

有关入口：**D1 IN STATUS**

D1 IN

D1 OUT STATUS

2.15 扩充的磁盘参数块

为了对不同格式的“外来”的一些磁盘能方便地进行读出和写入，与驱动器有关的所有参数都保持在RAM中，作为一个扩充的CP/M磁盘参数块（XPB），用户可以对一个XPB进行修改。

每一个驱动器有两个XPB。

XPB的格式为：

字节0……14：标准的CP/M DPB。

字节15：第一个扇区的编号。

字节16：每一个磁道的扇区数。

字节17：间隙长度（读/写的）。

字节18：间隙长度（格式的）。

字节19：为格式化所装填的字节。

字节20：$\log_2$（扇区尺寸）—7，upD765A的“N”。

字节21：扇区尺寸/128（必须是2的幂）。

字节22：保留用：当前磁道（由BIOS设置）

字节23：保留用：#00⇨不排列。

#FF⇨排列（由BIOS设置）。

字节24：#00⇨自动选择格式。

#FF⇨非自动选择格式。

对于一个特殊的磁盘，为了寻找XPB，则：

或者：调用BIOS SELDSK程序。它送回了DPH的地址、在DPH中，偏移量10处的字是DPB的地址。

或者：在# BE40 处一个字含有DPH矢量的地址。第一个DPH是驱动器A的，第二个DPH是驱动器B的。

或使用BDOS的功能31。

当调用SELDSK时，结果E的位0等于零，且XPB的字节24也等于零，则 BIOS 将企图决定磁盘的格式，并因此对XPB进行修改。

对不同的格式而言，XPB如下进行初始化：

· 系统格式：

36：　SPT，每一个磁道的记录数。

3：　BSH，块的移动。

7：　BLM，块的掩码。

0：　EXM，扩展掩码。

170：　DSM，块数－1。

63：　DRM，目录的入口数－1。

#C0：　AL0,2,目录块。

#00：　AL1

16：　CKS，校验和矢量的尺寸。

2：　OFF，保留的磁道。

#41　第一个扇区号

9　每一个磁道的扇区数。

42　间隙长度（读/写的）

82：　间隙长度（格式的）

#E9　装填的字节

2：　$\log_2$（扇区尺寸）－7

4：　每一个扇区的记录数

0：　当前磁道

0：　不排列

0：　自动选择格式。

· 纯数据格式：

36：　SPT，每一个磁道的记录数。

3：　BSH，块的移动。

7：　BLM，块的掩码。

0：　EXM，扩展的掩码。

179：　DSM，块数－1，

63：　DRM，目录入口数－1

#C0：　AL0,2，目录块

#00：　AL1

16：　CKS，校验和矢量的尺寸。

0：　OFF，保留的磁道。

#C1　第一个扇区号

9：　每一个磁道的扇区数。

42：　间隙长度（读/写的）。

82　间隙长度（格式的）

#E9　装填的字节

2　$\log_2$（扇区尺寸）—7

4　每一个扇区的记录数

0　当前磁道

0　非排列

0　进行自动选择格式。

IBM格式

32　SPT，每一个磁道的记录数。

3　BSH，块移动。

7　BLM，块的掩码。

0　EXM，扩展的掩码。

155　DSM，块数—1

63　DRM，目录入口数—1

#C0　AL0,2，目录的块数。

#00　AL1

16　CKS，校验和矢量的尺寸。

1　OFF，保留的磁道。

#01　第一个扇区号、

8　每一个磁道的扇区数。

42　间隔长度（读/写的）。

80　间隙长度（格式的）。

#E9　装填的字节

2　$\log_2$（扇区尺寸）—7。

4　每一个扇区的记录数。

0　当前磁道。

0　非排列。

0　自动选择格式。

2.16　重启指令和它们的安排

BIOS使用的重启指令以及它们的安排如下：

RST0：由CP/M使用。

RST1…RST5：由固件调用程序等用。

RST6：为用户可用——由DDT使用。

RST7：中断所使用。

一个应用程序可以毫不费力地使用RST 6。

一个应用程序不可使用RST7，因为它是专为中断所使用的，除非程序想阻止中断，则才可以使用RST7。

如果一个应用程序由于本身的目的希望使用重启指令1…5以及它门的安排，那么，它必须将进入到BIOS的所有入口全都阻断掉，并恢复重启的位置。

起先将所要求的重启地址复制到RAM中去、然后再进入到一个BIOS程序，并再退出，再根据RAM版本交换重启地址。

如果一个应用程序不直接地调用BIOS，那末，只有在#005的BDOS入口才需要被阻止。

重启地址在任何一个固件跳转块或扩充的BIOS跳转块程序被调用时，则也必须进行恢复。

所提供的DDT程序，对于使用RST6已经作过了参改，从而代替了常用的RST7。

2、17　替换寄存器组以及ENTER FIRMWARE程序的

使用。

固件为其本身之目的使用BC′和Carry′，并使其他替换寄存器和IY寄存器变得不确定。如果一个应用程序想使用这些寄存器，那末，BIOS将自动地保存用户的替换寄存器和 IY 寄存器（在进入到BIOS时，以及在退出时再恢复它门）。 在BIOS内部，这些寄存器是由固件根据要求来设置的。如果应用程序调用固件程序或扩充的 BIOS程序， 则它门必须通 过那些保存必要的寄存器的ENTFR FIRMWARE程序而被调用。

程序SET REG SAV E用来对设置或清除保存的寄存器进行选择。

如果保存的寄存器被使能，则一个应用程序便：

可以使用替换寄存器和IY寄存器；

可以在一个ROM下面保持堆栈；

必须使用 ENTER FIRMWARE 程序来调用固 件程序和扩充的 BIOS 程序。

（这是不推荐使用的，但例外的是。一个固件程序如果在用户和调用程序予先保存了BC′和Carry′，以及堆栈不在一个ROM下面时，则它可以被调用。 改变屏幕方或改变 ROM使能状态的程序将会改变BC′）。

如果寄存器的保存被禁止，则一个应用程序便：

必须不使用替换寄存器；

如果堆栈在一个ROM的下面， 则必须使用 ENTER FIRMWARE 程序；

如果堆栈不在一个ROM下面，则所以直接调用固件程序。

BIOS 将会使IY变得不确定。

在替换寄存器的保存处于使能情况下运行BIOS，以及应用程序按照上述规定进行工作，是竭力所推荐的使用方法。如果一个应用程序对寄存器的保存进行禁止，那么，当在退出时，应该重新再使能它。否则的话，某些标准的应用程序便不能工作。

SET REG SAVE 程序不得通过 ENTER FIPMWARE 来调用。

替换寄存器的使用方面的详细介绍，可以参阅固件手册。

注：SETUP 命令将寄存器的保存的使能称为“慢”，而对禁止则称为“快”。这种术语是容易使人误解的，它们在速度的差别是非常小的。

第三章 AMSDOS

AMSDOS 是 AMSTRAD CPC 464 计算机中为 DDI-1软盘接口所配置的磁盘操作系统。AMSDOS 能够使BASIC 程序以对磁带文件进行存取的相同的方式来存取磁盘文件，实际上，目前所用来存取磁带文件的一些程序只需作少量改动，或不需作改动，就可以用于存取磁盘文件。对于 AMSDOS来讲，不能兼容的主要原因是文件名的问题，文件的名称必须完全符含CP/M 的标准，所以，磁带文件的限制性要求得少得多。

AMSDOS 被设计成是用来补充CP/M，而不是与CP/M相对抗。AMSDOS 和 CP/M 都具在同样的文件结构形式，并且彼此之间可以进行读和写。

AMSDOS 与 CP/M BIOS一样也存放在ROM中。

3.1 特性

AMSDOS 提供了如下的方便性：

将磁带输入和输出数据流（＃9）与磁盘进行转换。因此，凡在磁带中可使用的方便性均可使用上磁盘中。

显示磁盘的目录。

清除磁盘文件。

更改磁盘文件的名称。

选择约定的驱动器和用户。

每当 AMSDOS 建立起一个新文件时，总是以＄＄＄的式样给出一个名称，而不管给定的名称。当文件处于关闭时，文件的任何一个原先的版本都用一个-BAK 来重新命名，而新的版本将＄＄＄重新命名为其相应的名称。任何现有的_BAK 版本则被删掉。这就给出了一种自动一级的文件的备用。

例如：如果磁盘含有 FRED_BAK 和 FRED_BAS 这两个文件，当用户发出 OPENOUT "FRED_BAS" 命令时，则 AMSDOS 将建立一个称之为 FRED_＄＄＄的新文件。当发出 CLOSOUT 命令时，现有的 FRED_BAK 文件就被删掉，而 FRED_BAS 则改名为 FPED_BAK，且 FRED_＄＄＄则改名为 FRED_BAS。

AMSDOS 的所有方便性都通过将磁带固件调用阻断掉，或通过一些外部命令来完成。

所阻断的固件调用有：

CAS IN OPEN

CAS IN CHAR

CAS IN DIRECT

CAS RETURN

CAS TEST EOF

CAS IN CLOSE

CAS IN ABANDON

CAS OUT OPEN

CAS OUT CHAR

CAS OUT DIRECT

CAS OUT CLOSE

CAS OUT ABANDON

CAS CATALOG

其余磁带固件调用程序则不受阻断，也不受影响。

AMSDOS 的外部命令有：

A　　选择约定的驱动器A；

B　　选择约定的驱动器B；

CPM　　冷引导CP/M

DIR　　显示磁盘的目录

DISC　　将磁带程序更改成磁盘的程序

DISC_IN 将磁带的输入程序改成磁盘的输入程序。

DISC_OUT 将磁带的输出程序改成磁盘的输出程序。

DRIVE　选择约定的驱动器。

ERA　　把文件删除掉

REN　　更改一个文件的名称

TAPE　　将磁带程序改成磁带的另一些程序

TAPE_IN 将磁带输入程序改为磁带另一些输入程序

TAPE¯OUT　将磁带输出程序 改为磁带 的另一些输出程序。

OSER　　选择约定的用户

所有来自 BASIC 的这些程序都必须在前面放一个“|”符号，有些还要求带有一些参数。

3.2　文件名

AMSDOS 的文件名与CP/M2.2的文件名向上兼容，在文件名中，在其名称和任何所嵌入的标点的前面和后面，可以允许规定用户以及允许含有无意义的空隔。

例如：

ANAME	缺少用户、驱动器和类型。
10：ANOTHER_BAS	缺少驱动器，指明用户数。
2A：WOMBAT_TXT	所有部分都指明。
-	缺少用户、驱动器和全部文件。
5B：POSSUM_$ $ $	含有空隔的一个文件名。
a:aard ? ark	小写字母，AMSDOS 会将它转换成大写字母。

用户号如果给出的话，应在0…15范围之内，驱动器必须用字母A和B。如果用户或驱动器被给定，则它们后面必须跟一个冒号。

下列一些字符可以在文件名和类型部分中使用：

字母，数字，！” # $ &' + - @ ↑ 、} {~

其他任何字符则会使命令变成无效，并出现如下的信息：

Bad COmmand

字符“？”和“*”都是通配符（见CP/M 操作系统手册）。

名称部分可以最长为8个字符。类型部分最长3个字符。

当分析一个文件名的时候，AMSDOS 会把小写字母转换成大写字符，并将位7移走。

如果省掉到用户或驱动器，那么，就假设当前的约定值。这些都是可以由用户来设置的。

如果类型部分被省略，则假设使用约定的类型。这取决于所使用的名称中的前后之间的关系，但是，一般情况下，都用三个空隔作为约定的类型部分。

3.3　文件头标

磁带的文件以2^K为一个块进行分割，每一个块都用一个头标作为开始。CP/M的一些文件却没有头标。AMSDOS的文件，有还是没有头标，取决于文件本身的内容。这对于以BASIC所编写的一些程序来讲不会产生什么问题，但是对磁带文件和磁盘文件有一个很重要的差异。

不保护的 ASCII 文件也没有头标。其他所有的AMSDOS文件，在文件的前面128个字节中都有一个头标，称为头标记录。这些头标是通过对记录的前面67个字节求校验和的方法来进行检查的。如果校验和与予期值相同，那么，头标是存在的；如果校验和不对，那末便认为没有头标存在。一个没有头标的文件可能会被错认为是一个有头标的文件，这是不希望的，但有时会可能的。

头标记录的格式如下：

字节0…63：　磁带的头标（见下）

字节64…66：　文件在字节上的长度，但不包括头标记录。这三个字节共有24位数，最低位的字节位于最低的地址。

字节67…68：　16位检验和的值，它是字节0…66的检验和。

字节69…127：　不使用。

由 AMSDOS 所使用的磁带头标如下：（关于磁带头标的使用可参阅主固件方面的资料）

文件名：

字节0：　　　　　用户号，#00…#0F。

字节1…8：　　　名称部分，填有空隔。

字节9…11：　　　类型部分，填有空隔。

字节12…15：　　　#00。

头标中的文件名是用来设立文件所用的文件名称。如果实际的名称或用户紧接着被改变，那末，该入口是不进行修改的。

块数	字节16	不使用，置成0。
最末块	字节17	不使用，置成0。
文件类型	字节18	按照盒式带。
数据长度	字节19…20	按照盒式带。
数据地址	字节21…22(*)	按照盒式带。
首块	字节23	置成#FF，仅用于输出文件。
逻辑长度	字节24…25	按照盒式带。
入口地址	字节26…27	按照盒式带。
不分配	字节28…63	按照盒式带。

(*)：原文字节19…20，似为误。

当一个没有头标文件为输入而被打开时，一个假头标就被建立在存储器中：

文件名	字节0	用户号，#00…#0F。
	字节1…8	名称部分，填有空隔。
	字节9…11	类型部分，填有空隔。
	字节12…15	0。
文件类型	字节18	#16，不保护的ASCII版本1。
数据位置	字节19…20	2K缓冲器的地址。

首块　　字节23　　#FF。

其余各个字节都置成为零。

3.4　磁盘的更换

在 AMSDOS 下，一个磁盘可以进行更换，或被拿走，不过，这是在驱动器没有进行存取时，以及在驱动器的输入文件没有打开时才可以这样做的。与CP/M 不同的是，不需要对一个磁盘进行“记入”（log in）

当一个磁盘正在进行写入时，若对磁盘进行更换的话，可能会破坏磁盘上的数据。

当一些文件处于打开时，若对磁盘进行更换的话，那末，AMSDOS 便立即对此进行检测，驱动器中所有打开的文件将被丢弃，并且产生出一个出错信息。尚未被写入的数据将被丢失，并且，最后一个目录的入口将写不到磁盘上去。然而，在对目录进行读出时，AMSDOS 才能检测到这种更换，这种读出是对文件的每16K进行的，并且不管一个文件是打开还是关闭的。因此，当有一些文件处于打开时，通过对磁盘的更换，有可能会使16K数据变得不确定了。

注意，如果对驱动器中的某些文件怀疑它是否打开，那么可以发出下面任何一个 BASIC 命令：CAT，RUN，LOAD，CHAIN，MERGE 或 CHAIN MERGE 而对输入文件和输出文件都进行丢弃。

3.5　磁带固件程序的阻断

当 AMSDOS 在初始化时，它就把相应的磁带跳转程序块的各个入口拷贝到它自己的数据区中。当选择DISC命令时，

磁带跳转程序块的各个入口由 AMSDOS 的各个入口写满，当选用 TAPE 命令时，便又恢复到原先的磁带各个入口上来。

在初始化时，磁盘程序被选择。

为了阻断跳转块的各个入口，应该遵照下面的操作顺序：将所要求的跳转程序块的入口的三个字节拷贝到你自己的数据区，至少这三个字节是什么，则不要任意设定。然后用你自己的 JMP RST 或随便什么命令去代替跳转块的入口。当你收到控制时，便恢复跳转块的入口，并对它进行调用。当你再一次收到控制时，就保存跳转块的入口，并用你自己的命令去替换它。这个顺序将不管跳转块入口的内容而工作。

注意：当阻断 AMSDOS 程序的时候，必须要依次执行上面的顺序，仅对跳转块入口进行拷贝是不能进行工作的，跳转块入口必须被恢复到它在跳转块中的原来的地方。

虽然在某些情况之下，AMSDOS 的程序和相应的磁带程序，其所返回的参数在解释上有所不同，但是，它们都含有相同的接口却是完全可能的。由磁带程序和磁盘程序所检测到的错误，都以进位位为假，和零位为假而返回。单是由磁盘程序所检测到有出错时，则以进位位为假，零位为真而返回，后面的一种情况与磁带程序的 BREAK条件相符合。在上述两种出错情况下，寄存器A存放着一个出错的编号。

当一个程序执行失败时(进位位为假)，则它在寄存器A中送入一个6位长的出错编号。如果出错的情况已经报告给用户了，则寄存器A的位6是零，位7是1。下面列出了出错的编号。

#0E　　文件未按予定所打开。

#0E　　文件的硬结束。

#1A　　文件的软结束。

#20　　命令不对，通常由文件名不正确引起。

#21　　文件已存在。

#22　　文件不存在。

#23　　目录已满。

#24　　磁盘已满。

#25　　磁盘已在文件打开时作过更换。

#26　　是一个只读型文件。

由软磁盘控制器检测到的错误，都以#40……#7F之间的个位有意义的值来反映，即位6一般为1，位7一般为0。其余各位含意如下：

位5　　数据错——数据或ID段有CRC错。

位4　　越时错。

位3　　驱动器未准备好。——在驱动器中没有磁盘。

位2　　没有数据。——找不到扇区。

位1　　不能进行写入——磁盘处于写保护。

位0　　地址标志丢失。

下面都是一些所阻断的程序的接口程序。

125　CAS IN OPEN　(DISC)　#BC77

为输入而打开一个文件

功能：为进行一个文件的读出而建立起读数据流，如果有的话，则对头标进行读出，原则，在存贮器中建立一个假头标。

入口条件：B存放文件名的长度。

HL放文件名的地址。

DE存放要使用的2K缓冲器的地址。

出口条件：如果文件打开正确，则：

进位位为真，

零位为假。

HL存放含有文件头标的一个缓冲器的地址。

DE存放数据位置（从头标算起）。

BC存放逻辑文件的长度（从头标算起）。

A存放了文件的类型（从头标算）。

如果数据流已经打开，则：

进位位为假，

零位为假。

A = #0E，（出错号）。

BC、DE 及 HL则不定。

如果文件因为任何其他原因而打不开，则：

进位位为假。

零位为真。

A为出错号。

BC、DE 及 HL均不定。

通常，IX 和其余标志位不定。

其中各寄存器保留。

注：所提供的2K缓冲器（2048个字节）是当对磁盘进行读出时，用来存放文件的内容的。一直到调用 CAS IN CLOSE 或 CAS IN ABANDON 而关闭文件时，缓冲器一直保持其使用。缓冲器可以位于存储器中任何的地方，甚至可以在一个 ROM 的低位部分。

文件名必须遵守 AMSDOS 中不用通配符的规定。文件名可以放在 RAM 中的任何地方，也可以放在一个 ROM 的低位部分中。

如果文件名的类型部分省略掉，则 AMSDOS 将试图依次地打开含有“—”，“_BAS”，“_BIN”等类型部分的一个文件，如果这些也不存在，则打开就失败。

当文件被打开时，文件的第一个记录立即被读出。如果该记录中含有一个头标，则头标就被拷贝到存储器，否则就在存储器中建立一个假头标，存放头标的存储区的地址送回给用户，因此，可以从这个区域中提取信息。这个存储区将位于 RAM 的中间32K中。用户不允许对头标进行写入，而只能对它进行读出。AMSDOS 为了其本身的一些目的，在头标中使用了一些段，这样，可以对从磁盘中所读出的内容区分开来，文件类型、逻辑长度、入口点以及所有的用户段将都保持不改变。

126 CAS IN COLSE（DISC） #BC7A

完全关闭输入文件。

功能：将读数据流关闭。

入口条件：无条件。

出口条件：如果数据流关闭正确，则

进位位为真，

零位为假，

A为不定。

如果数据流没有关闭，则：

进位位为假，

零位为假。

A = #0E（出错号）

如果由于某种原因，关闭不成功，则：

进位位为假。

零位为真

A = 出错号。

不管在什么情况下，BC、DE,HL和其他标志位均不定，其他各寄存器保留。

注：这个程序应该在用 CAS IN CHAR 或 CAS IN DIRECT 来读一个文件之后被调用，从而对这个文件进行关闭。

在调用这个程序之后，用户可以通过 CAS IN OPEN 而收回缓冲器。

在输入文件被关闭后，驱动器的马达立即进行关闭，这样就使得送到机器中的装入程序不致于因为马达正在运行而丢掉。

127 CAS IN ABANDON（DISC） #BC7D

立即关闭输入文件。

功能：从读数据流中放弃读出，并将它关闭掉。

入口条件：无条件。

出口条件：AF、BC、DE 及 HL 不定。

其他各寄存器保留。

注：这个程序主要在错误或相似情况出现之后时使用。

用户在调用这个程序之后可以通过 CAS IN OPEN 来收回缓冲器。

128 CAS IN CHAR（DISC） #BC80

从输入文件中读一个字符。

功能：从输入的数据流中读一个字符。

入口条件：无条件。

出口条件：如果字符读出成功，则：

进位位为真，零位为假。

A存放从文件中所读出的字符。

如果发现文件的结束点，或数据没有予期的得到打开，则：

进位位为假，零位为假，

A = #0E，#0F，#1A（出错号）。

如果由于某种原因执行不成功，则：

进位位为假，零位为真。

A = 出错号。

通常，IX和其他标志均为不定。其他各寄存器保留。

注：一旦第一个字符已经从一个文件中读出，则文件的剩余部分便可以一个字符，一个字符跟着读出（用 CAS IN CHAR）。若想用 CAS IN DIRECT 转换成直接读出，则是不可能的。

文件的CP/M 结束字符被处理为文件的结束点（进位位为假，零位为假），但是，继续对字符进行读出，一直到文件的硬结束点出现则是可能的。对文件的 CP/M结束，A寄存器置为 #1A，而对于文件的硬结束，则A寄存器置为 #0F。

置 #1A的这个动作不是由磁带程序完成的。为了对由其他程序所产生的CP/M 文件进行处理是必要的。如果已经知道文

件中含有二进制数据，那末将要求作一些特殊的动作，也就是说，文件的软结束点必须要忽略掉。当进位位为假的时候，版本1.0的磁带程序 CAS IN CHAR 绝不会送回 #1A 这一个数值的。

129　CAS IN DIRECT （DISC） #BC83

把输入文件读到存储器。

功能：直接将输入文件读到存储器，这个功能是一次完成的，而不是一个字符一个字符完成的。

入口条件：HL存放了放置文件所需的地址（在 RAM 的任何地方）。

出口条件：如果文件读出成功，则：

进位位为真，零位为假。

HL存放入口地址（从头标算起）。

A为不确定。

如果数据流没有予期打开，则：

进位位为假，零位为假。

A = #0E（出错号），

HL则为不确定。

如果由于某些原因，读出失败，则：

进位位为假，零位为真。

A = 出错号。

HL 则确定。

通常，BC，DE 和 IX以及其他标志位为不定。

其他各寄存器保留。

注：读数据流必须重新被打开（利用CAS IN OPEN）。

如果数据流已经为字符的存取所使用(利用了 CAS IN CHAR 或 CAS TEST EOF)，则直接地读这个文件就变成不可能的事了。多次地对文件直接读出也是不可能的。(任何企图这样做将破坏读出的文件的拷贝)。

如果文件具有一个头标，那么，所读出的字节数便是记录在24位的文件长度的这一段中的数(即磁盘文件头标字节64…66)，如果文件不含头标，那末这个文件一直被读到出现文件的硬结束点为止。

文件的CP/M结束字符 #1A，不处理为文件的结束点。

130 CAS RETURN (DISC) #BC86

将读出的最后一个字符放回。

功能：利用 CAS IN CHAR，将读出的最后一个字符放回到读出缓冲器中。这个字符将在下一次调用 CAS IN CHAR 时再度读出。

入口条件：无条件。

出口条件：所有寄存器及标志位均保留。

注：只能用这个程序将由 CAS IN CHAR 读出的最后一个字符送回，至少有一个字符已经一定被读出，原因是：

数据流已被打开，

或最后一个字符已被送回，

或已对文件的结束点作过最后一次测试。

131 CAS TEST EOF (DISC) #BC89

有没有已到达了输入文件的结束点?

功能：对输入文件有否已到达了结束点进行测试。

入口条件：无条件。

出口条件：如果末找到文件的结束点，则：

进位位为真，零位为假，A则不定。

如果已找到了文件的结束点，或数据流没有得到予期所打开，则：

进位位为假，零位为假，A＝#0F，#0F，#1A（出错号）

如果由于某种原因，程序执行失败，则：

进位位为假，零位为真，A＝出错号。

通常，IX 和其他标志位不定，其余各寄存器保留。

注：当在文件中没有字符了，或当读出的下一个字符是文件的CP/M 结束字符#1A时，这一个程序就告示文件为结束。

调用这个程序时，数据流处于字符输入方式。调用这个程序之后，再使用直接读出是不可能的事。

在这个程序已经被调用后，再去调用 CAS RETURN 也是不可能的，首先必须读出一个字符。

132　CAS OUT OPEN（DISC）　#BC8C

为输出而打开一个文件。

功能：为输出建立一个写数据流。

入口条件：B存放文件名的长度。

HL 存放文件名的地址。

DE 存放所使用的2K缓冲器的地址。

出口条件：如果文件打开成功，则：

进位位为真，零位为假。

HL 存放含有头标的一个缓冲器的地址。

A则为不定。

如果数据流已经打开，则：

进位位为假，零位为假。

A = #0E（出错号）

HL 则为不确定。

如果由于某种原因文件打开失败，则：

进位位为假，零位为真。

A = 出错号。

HL不确定。

通常，BC、DE、IX和其他标志位均不定。其他各寄存器保留。

注：当对文件使用 CAS OUT CHAR 而输出字符时，就由 AMSDOS 使用 2K 缓冲器来对输出进行缓冲。一直到用 CAS OUT CLOSE 或 CAS OUT ABANDON 来关闭文件时，才不使用缓冲器，缓冲器可以位于存储器的任何一个地方，甚至在低位 ROM中。

所送的文件名必须遵照AMSDOS的规定，不含有通配符。文件名被拷贝到写数据流的头标中去。文件名可以位于存储器的任何地方，甚至可以在一个 ROM 的低位部分。

文件以一个“_$$$”的类型部分被打开，而不管所提供的类型部分。任何具有同样名称和一个“_$$$”类型部分的现有文件都被删掉。文件重新改名它所提供的名称，这是在调用 CAS OUT CLOSE 时执行的。

当数据流被打开时，就建立起一个头标。在头标中的许多段都由 AMSDOS 设置，但剩下的部分都可由用户使用，头标的地址被送给用户，因此有些信息可以存放在它里面。用户可

以写入文件类型、逻辑长度、入口点以及所有的用户段。在头标中的其他任何段，用户不允许写入。除小文件类型是被置成非保护的 ASCⅡ（版本1）以外，用户可设置的段都全部初始化为零。对于字符输出，对头标类型段的写入应该在调用CAS OUT CHAR 之前完成，而且在这之后不应该再去作改动。

除非文件是非保护型的 ASCII 文件，否则，当文件在关闭时（CAS OUT CLOSE），头标将写在文件的开始部分上。

133　CAS OUT CLOSE（DISC）　#BC8F

完全关闭输出文件

功能：将写数据流关闭掉，并正确地给它一个名称。

入口条件：无条件。

出口条件：如果数据流关闭成功，则：

　　进位位为真。

　　零位为假。

　　A则不确定。

如果数据流未被关闭，则：

　　进位位为假，

　　零位为假，

　　A = #0E（出错号）。

如果由于某种原因，不能关闭，则：

　　进位位为假，

　　零位为真。

　　A = 出错号。

通对，BC、DE，HL，IX和其他标志位不定，其余各寄存器保留。

注：在被用 CAS OUT CHAR 或 CAS OUT DIRECT 之后，为了确保所有的数据都写到磁盘上去，也为了确保头标被写入到文件的开始以及对文件给出其真正的名称，必须调用这一个程序。

如果文件中未写入数据，那么，它就被丢弃，并且磁盘上什么也没有写入进去。这一点是与磁带程序相一致的。

当文件被打开时，它给出的类型部分是“_$$$”。这个程序将把文件的名称再改为其真正的名称，并且把任何现有的版本都改名为含有一个“_BAK”的类型部分。这保证了任何原先版本的文件都自动地保持为一个备用版本，而任何现有的“_BAK”版本都被删去，如果，当文件被打开时，调用人员不规定一个类型部分，那末 AMSDOS 对BASIC 文件将使用类型部分“_BAS”，对二进制文件用“_BIN”，对其他文件则用“一　”，作为对头标中的文件类型段的规定。

如果文件的实际长度不是128字节（一个CP/M 记录）的倍数，那么，一个CP/M的文件结束字符，#1A，就被加到文件上。这一个附加的字符是不记录在文件的长度上的。

如果写入作用被丢弃，则应该调用 CAS OUT ABANDON来作为对磁盘不再写入更多的数据。

在调用这个程序之后，用户可以利用CAS OUT OPEN来收回缓冲器。

134　CAS OUT ABANDON（DISC）　#BC92

立即关闭输出文件。

功能：丢弃输出文件并把写数据流关闭掉，任何没有写入的数据都作废，并且不再写入到磁盘上去。

入口条件：无条件。

出口件条：AF、BC、DE 和 HL不定。

其余所有寄存器保留。

注：这个程序主要在发生了一个错误等情况后才使用。

如果多于一个16k的物理区已经写入到磁盘，那末文件将以一个“_$$$”的类型部分出现在磁盘目录中，否则的话，文件将消失。这是因为每16k的一个文件就要求有一个目录。在16k都被写入后或文件被关闭时（CAS OUT CLOSE），才对磁盘写入一个目录。

135　CAS OUT CHAR（DISC）　#BC95

对输出文件写入一个字符。

功能：为了写数据流，对缓冲器加进一个字符。如果缓冲器已经满了，则在新的字符被输入前，缓冲器的内容被写到磁盘上。

入口条件：A存放了所要写入的字符。

出口条件：如果字符写入成功，则：

进位位为真，零位为假，

A为不确定。

如果数据流没有予期地打开，则：

进位位为假，零位为假。

A = #0E（出错号）

如果由于某种原因而失败，则：

进位位为假，零位为假。

A = 出错号

另外，IX和其他标志位不定，其余各寄存器保留。

注：在将所有字符都送到文件中之后，为了确保文件被正确地记到磁盘上，必须调用 CAS OUT CLOSE。
一旦本程序已被调用，就不可能再转换成直接对文件写（CAS OUT DIRECT）。

136　　CAS OUT DIRECT （DISC）　　#BC98

直接从存储器写输出文件。

功能：把存储器的内容直接写到输出文件上。

入口条件：HL存放要写的数据的地址（一直到头标）。
DE存放要写的数据的长度（一直到头标）。
BC存放入口地址（一直到头标）。
A存放文件的类型（一直到头标）。

出口条件：如果文件写入成功，则：
进位位为真，零位为假。
A则不定。
如果数据流没有予期地打开，则：
进位位为假，零位为假。
A＝#0E　（出错号）。
如果由于其他原因而失败，则：
进位位为假，零位为真。
A＝出错号。

通常，BC、DE、HL、IX和其他标志位均为不确定，其余各寄存器保留。

注：在写入文件之后，为了确保文件确实写入到磁盘，必须使用CAS OUT CLOSE来进行关闭。

把文件的写入方式由字符输出方式（用CAS OUT

CHAR）变成为直接输出方式（用CAS OUT DIRECT）是不可能的，反之也不行。（一旦方式已被选择之后）。

利用CAS OUT DIRECT不止一次地把文件直接写成两个或更多的部分也是不可能的。这将会写入不确定的数据。

137　CAS CATALOG（DISC）　#BC9B

显示磁盘的目录。

功能：显示当前驱动器和当前用户的磁盘目录。目录是以字母的次序存放的，并且是以许多列在当前的文本窗口（数据流＃0）上安放。在每一个文件的旁边，以K字节尺寸显示文件的大小。另外，也显示出磁盘上还可以使用的空间的总量。

入口条件：DE存放了所使用的2K缓冲器的地址。

出口条件：如果目录编制成功，则：

进位位为真，零位为假。

A则不定。

如果由于任何原因而失败，则：

进位位为假，零位为真。

A＝出错号。

通常，BC、DE、HL、IX和其他标志位不定，其余各寄存器保留。

注：标有SYS的各文件不显示。

标有R/O的各文件，在文件名后面用一个“☆”进行显示。

与磁带版本中的这一个程序不一样，不要求磁带/磁盘的输入数据流。（注意：当在编目录时，BASIC将输入的和输出的数据流都丢弃）。

3.6　外部命令

| CPM

功能：这一个命令关闭掉BASIC和AMSDOS，并冷引导CP/M。

参数：无。

注：使用MC STARTPROGRAM，因此所有的节拍链都丢失。

CP/M的实用命令AMSDOS_COM完成了这个命令相反的功能，而恢复了AMSDOS和BASIC。

| DISC.1N

功能：该命令把磁带输入固件的入口程序改成它们的磁盘的相应程序。

参数：无。

注：所修改的固件入口有：

CAS IN OPEN
CAS IN CLOSE
CAS IN ABANDON
CAS IN CHAR
CAS IN DIRECT
CAS RETURN
CAS TEST EOF
CAS CATALOG

I DISC.OUT

功能：该命令将磁带输出的固件入口改成它们的磁盘的对应程序入口。

参数：无。

注：所修改的固件入口有：

CAS OUT OPEN
CAS OUT CLOSE
CAS OUT ABANDON
CAS OUT CHAR
CAS OUT DIRECT

I DISC

功能：该命令将磁带的输入和输出固体入口都改成为磁盘用的相对应的入口。

参数：无。

注：所修改的固体入口有：

CAS IN OPEN
CAS IN CLOSE
CAS IN ABANDON
CAS IN CHAR
CAS IN DIRECT
CAS RETURN
CAS TEST EOF
CAS CATALOG
CAS OUT OPEN

CAS OUT CLOSE

CAS OUT ABANDON

CAS OUT CHAR

CAS OUT DIRECT

| DISC命令等效于| DISC.IN，| DISC.OUT两个命令。

| TAPE.IN

功能：该命令将磁带输入固件程序的入口恢复成AMSDOS进行初始化之前，它们所具有的状态。

参数：无。

注：所被恢复的固件的入口有：

CAS IN OPEN

CAS IN CLOSE

CAS IN ABANDON

CAS IN CHAR

CAS IN DIRECT

CAS RETURN

CAS TEST EOF

CAS CATALOG

| TAPE.OUT

功能：该命令将磁带输出的固件程序的入口恢复到AMSDOS进行初始化之前，他们所具有的状态。

参数：无。

注：所恢复的固件入口有：

CAS OUT OPEN

CAS OUT CLOSE
CAS OUT ABANDON
CAS OUT CHAR
CAS OUT DIRECT

| TAPE

功能：该命令恢复磁带固件入口到 AMSDOS 进行初始化之前的状态。

参数：无。

注：所恢复的固件入口有：

CAS IN OPEN
CAS IN CLCSE
CAS IN ABANDON
CAS IN CHAR
CAS IN DIRECT
CAS RETURN
CAS CATALOG
CAS TEST EOF
CAS OUT OPEN
CAS OUT CLOSE
CAS OUT ABANDON
CAS OUT CHAR
CAS OUT DIRECT

| TAPE命令等效于 | TAPE.IN和 | TAPE.OUT这两个令。

| A

功能：对驱动器A设置为约定的驱动器。

参数：无。

注：该命令相当于是用“A”作为参数的|DRIVE命令。

如果AMSDOS不能决定驱动器A中的磁盘的格式，那末，这个命令便失效。在这样的情况下，约定的驱动器原来是什么就保留什么。

当AMSDOS进行初始化时，约定的驱动器便是驱动器A。

| B

功能：将驱动器B设为约定的驱动器。

参数：无。

注：这个命令相当于是用“B”作为参数的|DRIVE 命令。

如果AMSDOS不能决定驱动器B中的磁盘的格式，那末该命令便失效。在这样的情况下，约定的驱动器不受改变。

当AMSDOS进行初始化时，约定的驱动器为驱动器A。

| DRIVE

功能：对当前的约定的驱动器进行设置。

参数：一个字符串参数。

注：字符串参数必须是一个在范围“A”…“P”或“a”…“p”内的单一字符。驱动器“C”…“p”是为以后增强时而用。

该命令如果不能决定所要求的驱动器中的磁盘的格式，则命令将失效。在这种情况下，约定的驱动器将保持不变。

当AMSDOS进行初始化时，约定的驱动器为驱动器A。

以BASIC给出的字符串参数，必须用地址形式送出。例如：

```
10A $ ="A"
20 |DRIVE, @A $
```

| USER

功能：设置一个约定的用户号。

参数：一个整数参数。

注：用户号必须在0…15范围之内。除此之外的任何一个其它参数都将造成出错，而约定的用户将保持不变。

当AMSDOS进行初始化时，约定的用户数为0。

| DIR

功能：显示磁盘的目录。

参数：一个可选的字符串参数。

注：参数是文件的名称，也可能包含通配符，只有那些与所给出的文件名相对应的文件才被显示出来。如果将参数省略掉，则假设为是“*.*”。

在磁盘上还可使用的空间的总数以K字节来表示。

显示在许多列上的目录将安排在当前的文本窗内（数据流＃0）。

凡标有SYS的那些文件则不显示。

没有一个

与CAS CATALOG（DISC）不同，目录既不存储也不显示其尺寸。其输出与CP/M DIR命令很相似。

以BASIC字符串给出的参数必须用地址送出，例如：

```
10 F $= "*.BAS"
20 |DIR, @F $
```

|ERA

功能：把文件消除掉。

参数：一个字符串参数。

注：字符串参数是一个文件名，也可能含有通配符。凡与这个文件名相应的所有文件均被清除掉。

虽与给出的文件名相应，但标有R/O的文件，则不被清除。此外，文件对一个16K就显示有一个信息。

如果，在磁盘上找不到一个与给出的文件名相应的文件，那末就显示一个出错信息。

以BASIC字符串给出的参数，必须用地址来送入，例如：

```
10F $= "FILE.BAS"
20 |ERA, @F $
```

|REN

功能：更改一个文件的名称。

参数：两个字符串参数。

注：第一个字符串参数是为文件所取的新名称。这个名称的文件必须是还不存在的文件。第二个字符串参数是被改名的文件的名称。

这两个名称均不能含有通配符。

这两个文件必须在同一个驱动器中。

这两个文件可以有不同的用户。

如果所要改名的文件标有R/O，那末将显示出一个出错信息，并且该文件不被改名。

所改过名的文件将有一个属性R/W DIR，而不管文件原来的属性是什么。

如果被改名的文件不存在，则显示出一个出错信息。

以BASIC字符串给出的参数必须用地址送出，例如：

```
10 OLDNAME $ = "OLDNAME.BAS"
20 NEWNAME $ = "NEWNAME.BAS"
30 | REN, @NEWNAME $, @OLDNAME $
```

3.7 AMSDOS 的一些信息

为了对磁盘进行存取，AMSDOS使用了CP/M BIOS。因此当磁盘出错时，将显示出BIOS的一些信息。这一节中说明了AMSDOS的一些信息的含义。

在下面，＜驱动器＞表示A或B。＜文件名＞表示了一个AMSDOS的文件名。

Bad command

在命令或文件名中，有一个语法错误。

＜文件名＞ already exists

用户想把一个文件的名称改为业已使用的名称。

＜文件名＞ not found

用户想对为输入而打开文件，消除一个文件，或改名一个文件，但这个文件是不存在的。

Drive <驱动器>： directory full

没有更多的空间可登录目录（每一个磁盘只能登录64个目录）。

Drive <驱动器> disc full

没有空间的碱盘记录块。

Drive <驱动器>： disc changed， closing <文件名>

当文件仍处于打开时，用户已经更换了磁盘。

<文件名> is read only

用户想一个标有只读型的文件进行消除或改名。如果文件目前的版本是属于只读型，想关闭这个文件，则也会产生这一个信息。

3.8 可用于AMSDOS中的BIOS的一些特性

AMSDOS利用CP/M BIOS来对磁盘进行存取。在AMSDOS下所运行的程序，为了可以直接对磁盘进行存取，可以利用 9个BIOS扩充的跳转块程序。

这些程序当作一些外部命令而取出。为了要找到所要求的程序的地址，可以调用 KL FIND COMMAND 。命令的名称都是一个单一的控制符（CtrlA…CtrlI）。

注* * BIOS 扩充的跳转块本身是不可使用的，实际上，它不存在一个AMSDOS的环境中* * 。

可使用的BIOS程序以及它们的命令名称如下所列：

SET MESSAGE (Ctrl A) (#01)
SETUP DISC (Ctrl B) (#02)
SELECT FORMAT (Ctrl C) (#03)
READ SECTOR (Cfrl D) (#04)
WRITE SECTOR (Ctrl E) (#05)
FORMAT TRACK (Ctrl F) (#06)
MOVE TRACK (Ctrl G) (#07)
GET DR STATUS (Ctrl H) (#08)
SET RETRY COUNT (Ctrl I) (#09)

这些程序的说明可参阅2.14节。

在 #BE40 处的一个字，存放了磁盘参数头标矢量的地址。磁盘参数头标矢量以及扩充的磁盘参数块可根据要求而进行修改。（见2.15节的详细说明）。

只有这里所提到的 BIOS特性才可以在AMSDOS下通过运行一个程序而使用。

3.9 存储器的要求

在对 AMSDOS进行初始化时，它在存储器中保留了 #500个字节的存储区，另外，内核也保留了四个字节为其外部命令的链接而用。

当使用AMSDOS程序CAS IN DIRECT将一个机器代码的程序从磁盘装入到存储器时， AMSDOS的 变量是不被写满的，这一点是很重要的。由于在一般情况上，变量在什么地方，是不可能看得出来的，所以这是一个问题,这是因为扩充的 ROM 的变量是动态地分配的。但注意，如果从磁带上将一个机器码程序送到存储器的话，就不会产生这样的问题。这是因为磁带

管理程序的变量是在固件的变量区中的。

AMSDOS在存储器的顶部保留了存储区，因此，最简单的解决办法是：把机器码程序装到存储器的底部。如果需要的话，程序可以把它自已再装到一个比较高的地址中去。

机器码程序可以分两个步骤进行装入：首先，将它装入，然后，再运行一个在存储器底部的装入程序。MC BOOT PROGRAM的作用是把所有的RSX均开闭掉，然后再对ROM进行扩展。此时，装入程序利用KL INIT BACK对AMSDOS进行初始化，因而迫使AMSDOS的变量进入你所希望的地方。这样，装入程序再与MC START PROGRAM一起使用AMSDOS的程序CAS OPENIN、CAR IN DIRECT和CAS IN CLOSE来把机器码程序进行装入。

为了利用KL INIT BACK来对AMSDOS进行初始化，则必须要求有AMSDOS的ROM号。为了决定AMSDOS的ROM号，必须对所选择的DISC程序检查所截断的磁带跳转块的入口。每个入口都是一个远调用，它的地址部分指在一个三字节的远地址处，远地址的第三个字节就是ROM号。在开闭AMSDOS之前，很明显，必须要这样做。

目前在磁带系统所开发的不含任何扩充ROM的机器码程序，仅仅只用来存放在#ABFF，以避免BASIC的变量。它们可以很容易地修改成使用AMSDOS。利用KL INIT BACK可以写入一些机器码来对AMSDOS进行初始化，与BASIC的使用情况几乎一样，ABSDOS将保留RAM在#ABFC的下面。

第四章 实用程序

磁盘系统提供了大量的实用程序。这其中,有些是 Digital Research公司提供的，它们都在资料SOFT159——CP/M指南中。其余部分则是由CPC464所专用的，即在下面要介绍的。

CPC464专用的实用程序有：

AMSDOS. COM	返回到AMSDOS和BASIC。
CLOAD. COM	把一个文件从磁带拷贝到磁盘。
CSAVE. COM	把一个文件从磁盘拷贝到磁带。
CHKDISC. COM	用两个驱动器对两个磁盘进行比较
DISCCHK. COM	用一个驱动器对两个磁盘进行比较
COPYDISC.COM	用两个驱动器从一个盘拷贝到另一个盘
DISCCOPY.COM	用一个驱动器从一个盘拷贝到另一个盘
FILECOPY.COM	拷贝一些文件。
FORMAT. COM	在驱动器A：中对一个盘进行格式化
BOOTGEN. COM	对磁盘写入引导程序和格式扇区。
SYSGEN. COM	对系统磁道写入CCP和BDOS。
SETUP.COM	在格式扇区中对数据进行改变。

4.1 AMSDOS. COM

介绍

AMSDOS 已经编写成允许用户从CP/M退出而回到 BASIC（在AMSDOS环境下）。实际上，AMSDOS 执行了一个与

|CPM相反的功能，|CPM是用来从AMSDOS退出而返回到CP/M。

AMSDOS. COM的使用

该程序通过送入“AMSDOS”而被调用。

该命令不需要参数，即使送入参数亦被忽略。

该程序在所有的前台ROM中搜索BASIC命令，当含有BASIC的ROM一旦找到，屏幕就进行清除，同时BASIC式样的信息被显示。现在，用户在BASIC. AMSDOS环境已被重新建立，这样，BASIC便可以使用磁盘系统。

如果未能找到含有BASIC的前台ROM，那末程序就显示出下面的信息：

BASIC not found

并执行一个热引导。

4.2 CLOAD. COM

介绍

CLOAD是一个CP/M过渡程序，它读一个磁带文件，然后把它写成一个磁盘文件。

CLOAD在对磁盘写入文件时，不建立AMSDOS的头标记录。这可以在使用AMSDOS来存取一个由CLOAD所建立的磁盘程序文件时，产生一些兼容性。

CLOAD的起动

CLOAD程序是存放在一个称之为CLOAD. COM的文件中的，因此可以用下面的命令进行调用：

CLOAD＜磁带文件＞，＜磁盘文件＞

CLOAD取两个命令参数，它们是在CLOAD命令的后面送入的，在参数的前面至少要有一个空隔符。两个命令参数之间可以用空隔或逗句分开。

第一个参数＜磁带文件＞，是所读出的磁带文件的名称，用引号（”）括起。如果不指明＜磁盘文件＞，那末，后面的引号可以省掉。如果不指明＜磁带文件＞，则CLOAD便读出磁带上所遇到的第一个文件。

如果＜磁带文件＞的第一个字符是“！”，那末，它就从文件名当中去掉，并且具有禁止由磁带读出转件所产生的一些信息的功能。

如果＜磁带文件＞的长度超过了16个字符，那末，便输出下面的信息，并废弃CLOAD的作用：

Invalid cassette filename

第二个参数，＜磁盘文件＞，是要写入的CP/M文件的名称。如果磁带文件也是一个有效的CP/M文件名，那末用户就不必再指明＜磁盘文件＞，在这种情况下，CP/M文件取同一个名称。用户不能在没有第一个指明的＜磁带文件＞的情况下面指明＜磁盘文件＞。

如果＜磁盘文件＞不是一个有效的CP/M文件，或者如果＜磁盘文件＞没有被指明，而磁带的文件名又不是一个有效的CP/M文件，那末，便显示下面的信息并废弃CLOAD的作用：

Invalid CP/M filename

所保护的一些文件

如果所读的磁带文件是受保护的，那末CLOAD便废弃，并

输出下面信息：

Cannot read protected Cassette files

磁盘文件的命名

当CLOAD将数据从磁带传送到磁盘时，便以。$$$作为类型而建立一个临时的文件，在传送全部完成之后，如果所指明的文件已经存在，则它就以BAK作为其类型而重新命名。

其他信息

program error：Cassette stream in use

当CLOAD要打开磁带文件时，由于当前的数据流正在使用而使打开失败，那末便出现这个信息。这个信息在一般情况下是不应该产生的。

program error： Cassette strcam not in use

当CLOAD要关闭磁带文件时，由于当前的数据流不在使用中，而使关闭失败，则就出现这个信息。通常情况下，不应该产生这个信息。

* * Break* *

如果【ESC】键在从磁带上读出文件时按下，则便输出这个信息。

Disc dirctory full

当磁盘的目录已经满时，就输出这个信息。

Failed rename temporary file

如果CLOAD不能将临时文件重新取名为所指定的磁盘文件名，则出现这个信息。一般情况是不应该出现这个信息的。

Disc or directory full

如果由于磁盘或目录已经满了，而不能使CLOAD对磁盘写入一个记录，则输出这个信息。

Failed to close CP/M File correctly

当CLOAD关闭CP/M文件时，产生了一个错误，则输出这一个信息。

4.3 CSAVE. CQM

介绍

CSAVE是一个CP/M的过渡程序，它用来读出一个磁盘文件，然后再将这个文件写入到磁带文件上。

CSAVE不使用对磁盘写入一些文件时由AMSDOS所建立起来的头标。所以，当使用CSAVE来存取一个由AMSDOS所建立起来的磁盘文件时，可以有一些兼容性。

当CSAVE打开磁盘文件时，它将文件类型设为是一个ASCII文件版本1。

CSAVE的起动

CSAVE程序是放在一个名为CSAVE. COM的文件中，它可以用下面的命令来进行调用：

CSAVE <磁盘文件>，<磁带文件>，<写入速度>

CSAVE用了三个命令参数，它们都是有CSAVE命令的后面输入的，在参数的前面至少要有一个空隔符。命令参数之间可以用空隔或逗号给以分开。为了指明特定的一个参数，所有在其前面的参数则也必须指明。

＜磁盘文件＞是所读的CP/M文件的名称。用户必须要送入这个参数。文件名中不得含有任何一种通配符（*或？），如果＜磁盘文件＞是一个无效的CP/M文件，那末CSAVE就被废弃，并且出现下面的信息：

Invalid CP/M filename

＜磁带文件＞，如果有的话，指明了要写入的磁带文件的名称，它用双引号（”）括起来。如果在＜磁带文件＞的后面不再给出参数，那末最后一个双引号可以省掉。如果＜磁带文件＞不给出，那末CSAVE便使用CP/M的文件名。

如果＜磁带文件＞的第一个字符是“！”，那末它就从文件名中去掉，并具有禁止由磁带读出软件所产生的一些信息的功能。

如果＜磁带文件＞超过了16个字符，那末就出现下面的信息，并且废弃CSAVE的作用：

Invalid ca ssette filename

＜写入速度＞，如果有的话，用来选择数据写入到磁带中去的两种速度中的其中一种。零表示用每秒钟1000 bit的速度，而1表示用每秒钟2000 bit的速度。如果＜写入速度＞不给出，则就设定为用1000bit/秒的速度。

其他一些信息

Program error：Cassette stream in use

如果由于当前的数据流在使用着而使 CSAVE不能打开磁带文件，则输出此信息。在一般情况下，不应该产生此信息。

Program error：CassetteIstream not in use

如果由于当前的数据流不在使用中而使 CSAVE 不能关闭磁带文件时，则输出此信息。一般情况下，不应该产生此信息。

Invalid speed setting

（You may only Specify 0 or 1）

如果用户规定了一个非法的速度设置值，则输出此信息。

* * Break* *

如果对磁带带进行读出时，按了【ESC】键，则输出此信息。

CP/M file does not exist

如果要读出的CP/文件不存在，则输出此信息。

4.4 CHKDISC．COM

介绍

CHKDISC是一个CP/M过渡程序，它用来检查驱动器B中的磁盘是否是驱动器A中的磁盘的完全的复制品。该程序提供出三种AMSTRAD CP/M的磁盘格式。

CHKDISC 是为在两个驱动器系统而用，对于一个驱动器系统，则使用DISCCHK。

CHKDISC要求最小的TPA的尺寸为39K。

CHKDISC 的起动

CHKDISC程序是存放在一个名为CHKDISC. COM 的文件当中，并可以CHKDISC命令进行调用，此时，会输出下面的提示：

Please insert source disc into drive A
and destination disc into drive B
Then press any key

源盘就是原始的磁盘，目的盘那是所假设的拷贝。

磁盘格式

用户把磁盘插进驱动器A和B之后，CHKDISC 便建立了源盘的格式。该格式与目的盘的格式进行比较。如果两个盘的格式不一样，或目的盘还未进行格式化过，那末，CHKDISC 便显示下面的信息，并且该程序被废弃：

Source and destination dlscs have
different formats

源盘就是原始的磁盘，目的盘即是所假设的拷贝。
否则，以下面的信息而开始进行检查：

Copy Checking started

并且以下面的信息的出现而结束检查：

Copy checking complete

拷贝的检查

两个磁盘的比较从磁道 0 开始，源盘上的每一个磁道与目的

盘中的数据进行比较。如果发现数据不相对应，那末就显示下面的信息：

Failed to verify desstination disc

COrrectly：track n sector m

这里，n和m分别表示坏的扇区的磁道号和扇区号。

多重检查

当检查完成时，便发出下面的提示：

Do you want to check another disc （Y/N）：_

要想对另一个磁盘进行检查，用户应该送入Y。如果想退出CHKDISC并返回到CP/M，则用户应该送入N，结果，发出下面的提示：

Please insert a CP/M system disc into drive A

then press any key：_

此时，用户应该把一个CP/M系统插入到驱动器A中去，然后按任何一个键。这样，CHKDISC便开始进行一个热引导，从而返回到CP/M。

其它一些信息

you must insert the source disc into A

如果当用户已被提示需插入磁盘时，在驱动器A中还没有磁盘，那末就输出这一个信息。此信息给出之后，相当于再一次提醒用户，将源盘插入到驱动器A中去。

you must insert a CP/M system disc into

drive A

当已经提示用户插入一个CP/M系统的磁盘之后，如果驱动器A中还没有磁盘，就输出这个信息。此后，相当再一次提醒用户，把一个CP/M系统磁盘拖入到驱动器A中。

you must insert the destination disc into drive B

当已向用户提示要插一个盘时，如果在驱动器B中还没有磁盘，那末就输出些信息。输出此信息后，相当再一次提示用户把目的盘插入到驱动器B中去。

WARNING：Failed to compare disc correctly

当进行拷贝时，如果CHKDISC程序被废弃，则出现此信息。

The source disc has a unknown format

如果CHKDISC程序对源盘的格式不能识别，则出现此信息。输出此信息之后，CHKDISC即被废弃。

Reading track：n

这里，n是当前的磁道号。当CHKDISC程序正在对源盘读某一个磁盘时，显示此信息。

Checking track： n

这里，n是当前的磁道号。当CHKDISC程序正在对目的盘的某一磁道进行比较时，显示此信息。

Failed to read source disc correctly：

track n sector m

这里，n 和 m 分别是坏扇区的磁道号和扇区号。如果 CHKDISC 程序对源盘读一个扇区读得不正确，那就出现此信息。显示此信息之后，该CHKDISC程序被废弃。

Failed to verify destination disc correctly:

track n sector m

这里，n和M分别为坏扇区的磁道号和扇区号。当从目的盘中读回的数据与源盘中的数据不一致时，显示此信息。

↑C…aborted

当CHKDISC正在为用户的输入而等待时，如果用户欲按了一个control_c，则显示此信息，之后，CHKDISC 程序被废弃。

Illegal message number: program aborted

该信息表示了在CHKDISC程序的执行中有一个错误。通常情况下，不应该有这样的情况。

Insufficient space in TPA

如果包含有CHKDISC所使用的缓冲器的过渡程序区的尺寸大小，则显示此信息。用户应该用 39^K或更大一些的CP/M 系统盘进行热引导，并再一次调用CIIKDISC。

4.5 DISCCHK. COM

介绍 DISCCHK是一个CP/M过渡程序，它是利用一个磁盘驱动器，检查一下一个磁盘上的内容是否是另一个盘上的内容

的真正的拷贝。该程序支持有三个AMSTRAD CP/M的磁盘格式DISCCHK用于一个驱动器系统，CHKDISC 则用于两个驱动器系统。

DISCCHK一次检查8个磁道，并要求至少有一个 39^K大小的TPA。

DISCCHK的起动

DISCCHK程序放在一个名为DISCCK.COM的文件中，并可以用DISCCHK命令来调用它。此时，显示下面的提示：

please insert source disc into drive A

then press Any key

当检查开始时以及结束时，就显示下面的信息：

copy checking started

copy checking complete

磁盘格式

在用户已经将源盘插入驱动器A之后，DISCCHK程序就建立起它的格式。这个格式与目的盘的格式相比较。如果两者不一样，或者目的盘还未格化式过，那末DISCCHK 程序被废弃并显示下列信息：

Source and destination discs have

different formats

拷贝的检查

两个磁盘一次比较 8 个磁道，从0磁道开始检查起。八个磁道的每一个块都从源盘中进行读出，然后与目的盘上的数据相

比较。如果发现数据不一致，则显示下列的信息：

Failed to verify destination disc correctly:
track n sector m

这里，n和m分别表示坏扇区中的磁道号和扇区号。

换盘

DISCCHK只用驱器动A来进行检查。在对源盘进行读出之前，显示下列提示信息：

please Insert source disc into drive A
then press any key: _

在检查目的盘之前，则显示下面的提示：

please Insert destination disc into drive A
then press any key: _

用户必须保证将正确的磁盘插入到驱动器A中。

多重检查

当拷贝检查完毕后，就发生下面的提示：

Do you want to check another disc (Y/N): __

要想对另一个磁盘进行检查，用户应该送入Y。要想退出DISCCHK并返回到CP/M，用户应该送入N，其结果是发出如下提示：

please insert a CP/M system disc into drive A
then press any key: _

用户此时就应该把一个CP/M系统盘指到驱动器A中去，然后再按一个键。DISCCHK然即就开始一个热引导以返回到CP/M。

其他一些信息

You must insert the source disc into drive A

当已经提示用户需将磁盘插进驱动器 A后，驱动器A中还没有磁盘时，则显示该信息。此后，再向用户提示，将源盘插入入到驱动器A中去。

You must inerst a CP/M system disc into drive A

当已提示用户需将一个CP/M系统盘插进驱动器A后，驱 动器A中还没有磁盘时，就显示此信息。之后，再向用户提示，将一个CP/M系统盘插入到驱动器A中去。

You must insert the destination disc into drive A

在向提示需在驱动器A中插入一个磁盘后,驱动器A中还没有磁盘时，就发生该封息，在此之后，用户被再次提示将目的盘插入到驱动器A中去。

WARNING：Failed to compare disc correctly

当进行拷贝时，DISCCHK程序被废弃，则发出此信息。

The source disc has an unknown format

如果DISCCHK不能识别源盘的格式，则显示出该信息。 该信息输出之后，DISCCHK程序被废弃。

Source and destinatlon discs have different format

源盘与目的盘的格式不相同时，显示该信息。输出此信息后，DISCCHK程序被废弃。

Reading track： n

这里，n是当前的磁道号，当DISCCHK从源盘上读一个磁道时，显示此信息。

Checking track： n

这里，n是当前的磁道号，当DISCCHK正在对目的盘的某一磁道进行比较时，显示该信息。

Failed to read source disc correctly：

track n sector m

这里， n 和 m 分别表示了坏扇区的磁道号及扇区号， 当DISCCHK从源盘上读错时，显示此信息。输出该信息后，DISCCHK被废弃。

Failed to read destination disc correctiy：

track n sector m

这里， n 和 M 分别表示了坏扇区的磁道号和扇区号。如果DISCCHK 不能从目的盘中读到正确的数据，便显示该信息。输出此信息后，DISSHK被废弃

Failed to verify destination disc correctrly：

track n rector m

这里，n和m分别为坏扇区的磁道号和扇区号。如果从目的

盘中读回的数据与源盘中的数据不一致，则显示该信息。

↑C…abopted

如果DISCCHK正在为用户输入而等待时，用户按下了Contrcl_c键，那末就显示此信息。该信息显示之后，DISCCHK被废弃。

Illegal message number: program aborted

表示了在执行DISCCHK程序的过程中有了错误。一般，这信息不应该产生，

Insufficient space in TPA

由DISCCHK所使用的含有缓冲器的过渡程序区太了。用户应该用一个39K或更大一些的CP/M系统盘进行热引导，同时再一次调用DISCCHK。

4.6 COPYDISC.COM

介绍

COPYDISC是一个CP/M过渡程序，其功能是将驱动器A中的磁盘上的数据拷贝到驱动器B中的一个磁盘上。程序支持有三种AMSTRAD CP/M磁盘格式。

COPYDISC用于两个驱动器系统，DISCCOPY则用于一个驱动器系统。

COPYDISC至少要求有39K大小的TPA。

COPYDISC的起动

COPYDISC程序存放在一个名日COPYDISC.COM的文件

中，并且可以用命令COPYDISC来调用这个程序。调用之后，给出下面的提示信息：

please insert source disc into drive A
and destination disc into drive B
then press any key

当拷贝开始进行以及结束时，显示出下列信息：

copying started
copying complete

磁盘的格式

用户已将磁盘插进驱动器A和B之后，COPYDISC 程序就建立起源盘的格式。该格式与目的盘的格式相比较，如果两者不一样，或目的盘还未被格式化过，那末，COPYDISC将对目的盘的每一磁道进行格式化。

拷贝

源盘从磁道0开始进行拷贝，写入到目的盘每个磁道的内容，利用把它读回，并与原来的数据进行比较的方法来校验拷贝的正确性。

多重拷贝

当拷贝完成之后，给出下面的提示：

Do you want to copy another disc（Y/N）：_

要想对另一个磁盘进行拷贝，则用户需按下Y。要想退出COPYDISC程序，并返回到CP/M，则用户应该送入N，并在

此时得到提示如下：

please insert a CP/M System disc into drive A

then press any key：_

此时，用户应该在驱动器A中插进一个CP/M系统磁盘，并任任意按一个键。此时，COPYDISC就开始一个热引导以返回到CP/M。

其他一些信息

You must insert the source disc into drive A

当用户被提示要在驱动器A中插入一个磁盘时，驱动器部没有磁盘，则输出此信息。此信息输出之后，用户再被提示将源盘插入到驱动器A。

You must insert a CP/M system disc
into drive A

当用户已被提示要在驱动器A中插入一个CP/M系统盘时，驱动器A中还没有磁盘，那末便发出此信息。该信息发出之后，用户再次被提示将一个CP/M系统盘插进驱动器A。

You must insert the dest!nation disc into
drive B

当用户已被提示要把一个盘插进驱动器B时，驱动器还没有磁盘，则发出此信息。发出这个信息就是提示用户，把目的盘插进驱动器B中去。

The destination disc in drive B must be

write-enabled

如果驱动器B中的盘是受到写保护时，便输出此信息。这个信息再次提示用户，把目的盘插进驱动器B。

Formatting whilst copying

如果目的盘末被格式化过，或它与源盘的格式不一样时，输出此信息。

WARNING: Failed to copy disc correctly
The destination disc not be used until
it is successfully copied onto

当正在拷贝时，如果COPYDISC程序被废弃时，输出该信息。

The source disc has an unknown format

如果COPYDISC程序不能识别源盘的格式，则输出此信息。并使该程序被废弃。

Reading track: n

当COPYDISC程序正从源盘上读出一个磁道时，便输出该信息。其中n是当前的磁道号。

Writing track: n

当COPYDISC程序正在对目的盘的磁道进行写入或检验时，输出该信息。其中n为当前的磁道号。

Failed to read source disc correctly:

track n sector m

这里n和m分别是坏扇区的磁道号和扇区号。如果COPYDISC程序从源盘上对一个扇区进行读出时，结果不正确，则输出此信息，并且该程序亦被废弃。

Failed to write destination disc corretly:

track n sector m

这里，n 和 m 分别是坏扇区的磁道 号 和 扇 区 号。 如 果COPYDISC程序对目的盘的一个扇区写入不正确，则此信息被输出之后，此程序被废弃。

Failed to read destination disc dorrectly:

track n sector m

这里n和m分别是坏扇区的磁道号和扇区号。如果COPYDISC程序从目的盘的一个扇区读出时，读得不正确，则 输 出 此 信息。之后，此程序被弃废。

Failed to verify destination disc correctly:

track n sector m

这里n 和M分别是坏扇区的磁道号和扇区号。如果从目的盘中读回的数据与源盘的数据不一致时，输出此信息，

Failed to format destion disc correctly:

track n

这里n是当前的磁道号。如果COPYDISC不能正确地对 目的

盘的一个磁道进行格式化，则输出此信息。之后，该程序被废弃。

↑C…aborted

如果 COPYDISC 程序正在等待用户的输入时，用户按了Control-c，则输出该信息，且该程序被废弃。

Illegal message number: program aborted

该信息表示了在COPYDISC 程序的执行过程中有了错误产生。一般情况上，不应该出现该信息。

Insufficfent space in TPA

如果含有BOPYDISC所使用的缓冲器的过渡程序区的尺寸大小，但输出该信息。用户此时应该用一个39^K的，或是更大的CP/M系统盘进行热引导，并再一次调用COPYDISC程序。

4.7 DISCCOPY.COM

介绍

DISCCOPY是一个过渡程序，它使用一个盘驱动器，把一个磁盘上的数据拷贝到另一个磁盘上。该程序支持了三种AMSTRAD CP/M磁盘的格式。

DISCCOPY是用于一个盘驱动器系统，COPYDISC则是使用于两个盘驱动器系统中。

DISCCOPY一次拷贝8个磁道，它要求TPA最小的尺寸为39^K。

DISCCOPY 的启动

DISCCOPY程序存放在一个名为DISCCOPY.COM的文件中，并可以用DISCCOPY命令来对它进行调用。调用后，输出下面的提示给用户：

please insert source disc into drive A then press any key

在拷贝开始或结束时，又显示下列提示：

copying started

copying complete

磁盘的格式

当用户把源盘插进驱动器A后，DISCCOPY便对它建立格式。这个格式与目的盘的格式进行比较。若两者不一致，或目的盘还未格式化过，则此程序将对目的盘的每个磁道进行格式化。

拷贝

源盘一次拷贝8个磁道，从磁道0开始。8个磁道中的每一个块被读出，然后再写到目的盘下去，并进行校验，一次一个磁道地进行。

磁盘的更换

DISCCOPY程序只用磁动器A来进行拷贝。在从源盘读出之前，先显示下面的提示：

please insert source disc into drive A

then press any key

在对目的盘写入之前，先显示下面的提示：

please insert destination disc into drive A

then press any key

用户必须保证把正确的磁盘插到驱动器A。

多重磁贝

当拷贝结束后，发出下面的提示：

Do yot want to copy another disc（Y/N）：_

若要拷贝另一个盘，用户应按Y。若要退出DISCCOPY并返回到CP/M，用户则应该送入N，其结果是又发出提示：

please insert a CP/M system disc into drive A

then press any key：_

此时，用户应该在驱动器A中插入一个CP/M系统盘，并再按一个键。DISCCOPY则开始一个热引导以返回到CP/M。

其他一些信息

you must insert the source disc into drive A

如果已提示过用户把磁盘插进驱动器A，但驱动器A还没有磁盘，则输出该信息。之后，用户再一次被提示，将源盘插进驱动器A。

you must insert the destination disc into drive A

如果用户已被提示过把磁盘插进驱动器A，驱动器A还没有磁盘时，则输出此信息。此信息再一次提示用户，将目的盘插到驱动器A。

you must insert a CP/M system disc into drive A

如果用户已被提示过把一个CP/M系统盘插进驱动器A，但驱动器A中还没有磁盘时，输出此信息。此信息再一次提示用户，把一个CP/M系统盘插到驱动器A。

The destination disc in drive A must be
write-enabled

如果驱动器A中的磁盘是写保护的，则输出此信息。此信息输出后，用户被再一次提示，把目的盘插进驱动器A。

Format whilst copying

如果目的盘未格式化过，或它与源盘的格式不一致时，则输出此信息。

WARNING: Failed to copy dise correctly
The destination dise should not be used until
it is successfully copied on to

如果DISCCOPY程序在拷贝过程中被废弃，则产生这个信息。

Reading track: n

当DISCCOPY程序正从源盘上的某一磁道读数据时，输出此信息。这里的n表示当前的磁道号。

writing track: n

当DISCCOPY 程序为目的盘的某一磁道正在进行写入或校

验时，输出这个信息。这里的n表示了当前的磁道号。

Failed to read source dise correctly:
track n sector m

这里n和M分别为坏扇区的磁道号和扇区号。当DISCCOPY程序从源盘中读一个扇区时，读出不正确，则输出此信息。并使程序废弃。

Failed to write destination disc correctly:
track n sector m

这里，n和M分别指坏扇区的磁道号和扇区号。当DISCCOPY程序对目的盘的一个扇区写入时，不正确，则输出该信息。并使该程序废弃。

Failed to verify destination dise correctly:
track n sector m

这里，n和M分别指坏扇区的磁道号和扇区号。如果从目的盘所读回的数据与源盘不一致，则输出此信息。

Failed to fomat destination disc correctly:
track n

这里的n系指当前的磁道号。如果DISCCOPY目的盘的一个磁道格式化得不正确，则输出此信息。并使该程序被废弃。

↑C…aborted

如果DISCCOPY正在等待用户的输入时，用户送入了

Control-c，则输出此信息。并使该程序被废弃。

Illegal message number： program aborted

该信息表示DISCCOPY程序在执行中有错。一般不应该如此。

Insufficient space in TPA

如如含有DISCCOPY所使用的缓冲器的过渡程序区过小，则输出此封息。用户应该用一个39^K或更大的CP/M系统盘进行热引导，并再一次调用DISCCOPY程序。

4.8 FILECOPY.COM

介绍

FILECOPY 的功能是：当在不同的用户区之间任意传送一些文件时，从一个磁盘，把文件拷贝到另一个磁盘。所有的拷贝作用是用驱动器来实现的。如果使用一个双驱动器系统，则建议用PIP来进行文件的拷贝。三种AMSTRAD CP/M 盘的格式均支持，但是，所写入的磁盘必须是格式化好的。源盘和目的盘可以是同一个盘。

所拷贝的文件，在被写入之前先读到一个缓冲器。所用的缓冲器的存储量是动态地选择的，以满足TPA所要用的尺寸。

任何具有只读状态（R/O）的源文件将以读/写（R/W）而拷贝到目的盘上。目录中凡隐含有SYS 属性的源文件都不能进行拷贝。

任何时候，用户利用Control-C 进行输入时，程序则可能被废弃。目的盘下具有相同名称的任何一个文件被拷贝时，将引起复盖作用。因此，用户应该确保有价值的一些文件不要弄

丢掉。

FILECOPY的起动

用下面的命令，可以调用FILECOPY程序：

FILECOPY <filename> <1users>

FILECOPY可以给出一个或两个命令参数，它们都在该命令的后面被送入，在命令与参数之间必须有一个或几个空隔符。第二个参数必须用至少一个空隔符与第一个参数隔开。如果不给出参数，则有下面的信息出现：

NO SOURCE file Present on input Line

然后，该命令就被废弃掉。

第一个参数，<filename>，是要拷贝的磁盘文件的名称，它必须符合CP/M 的约定。在<filename>参数中，可以使用通配符"？"和"*"。在这种情况下，凡是与之相应的所有文件都将进行拷贝。

第二个参数，</users>，是可选的。如果这个参数被使用的话，则第一个字符"／"必须要用上，否则参数将被FIL-ECOPY所忽略。参数</users>允许选择所读的文件的用户区，以及所写入的文件的用户区。它包含了一个或两个的选择，可以以任意的次序来规定。在"／"字符后面，不允许有空隔符，若有一个空隔符,那末所有的选择都将被FILECOPY忽略掉。

所读出的用户区，那为源用户，可以用"S"后面跟着用户号来选择。

所写入的用户区，即为目的用户，可以用"D"后面跟用户号进行选择。

所选择的这两个用户号必须是 0…15 范围之内的十进制数，否则，就显示下面的信息并使程序废弃：

Invalid user number in options

用户号若小于10，则在用户号前面可以加一个0。

如果FILECOPY不能识别任何一种选择，或如果用户号大于两位数，则将显示下面的信息，并使程序废弃：

Syntax error in options

当使用“S”选择，则显示出：

COPYing will be from user n

这里，n是一个0…15之内的十进制数，并显示出来，以便确认。

当使用“D”选择时，则显示：

COPYing will be to user n

如果“S”和“D”都选用，则显示：

COPYing will be from user n to user n

如果这两个参数都省掉，或都被FILECOPY所忽略，则当前用户号被假设为进行文件的读出和写入，并不予显示确认。

单个文件的拷贝

当<filename>中不包含任何通配符时，说明只拷贝一个文件，此时，给出下面的提示：

Please insert SOURCE disc into drive A

then press any key：—

此时，用户应该把含用要拷贝的文件的磁盘插进驱动器，

然后再按任何一个字母-数字键。从而FILECOPY将在该磁盘上寻找源文件。如果找不到文件，则给出信息：

NO SOURCE file present on disc

并且FILECOPY会提示你把一个CP/M系统盘选入驱动器，然后该程序被废弃。

当源文件已被找到，则给出信息：

COPYing started……

同时，把文件读到一个缓冲器中，一直持续到文件结束或缓冲器已装满。然后又提示：

Please insert DESTINTION disc into drive A

then press any key：—

此时，用户应该把文件被写入的盘插进驱动器A，然后再按一个键。因此，缓冲器就对磁盘进行写入。如果文件一次不能拷完，则会再次向源盘提出请求，重复上述的拷贝过程，一直到目的文件拷完时才完毕。应注意，一个大的文件可能需要更换好几个磁盘。在完成拷贝之后，显示下面的信息：

＜文件名＞ Copied。

给以确认。其中，＜文件名＞将是实际上所拷贝的文件的名称。

当全部文件都已拷贝完时，会输出下面的信息：

COPYing Complete

Please insert a CP/M system disc into drive A

then Press any key：—

此时，用户应将一个含有CP/M 的磁盘插进驱动器，并按一个键，FILECOPY才完全结束。

多个文件的拷贝

当<文件名>含有通配符时，FILECOPY 程序将允许与其相应的任何文件都进行拷贝，并以一个连续的方式进行。在为源盘作了提示之后，便显示下面的信息：

Ambiguous filename：

Confirm individaul files（Y/N）？

如果用户想只拷贝一些所相应的文件，则送入“Y”，在此种情况下，FILECOPY将对所用相应的文件名进行搜索，并在屏幕上一一给出每个文件的名称，并显示：

<文件名> COPY（Y/N）？

这里，<文件名>即是相应的文件的名称。如果用户送入“Y”，则文件名就被存放在一个缓冲器里，准备进行拷贝，并继续搜索相应的其他一些文件。如果对各个文件不选择需要确认，那末，所有相应的文件不进行列出，而被读到一个缓冲器里，如果FIECOPY在源盘上找不到任何一个相应的文件名，那末就给出：

NO SOURCE file present on disc

同时，FILECOPY程序将被废弃。

如果用户送入“N”，那末将显示：

NO files to COPY

同时，FILECOPY程序被废弃。

一旦所有相应的文件名都已被得到，那末，每一个文件就如进行单个文件拷贝那样而进行拷贝。

几个应用例子

FILECOPY STAT。COM

对单个文件STAT。COM进行拷贝。

FILECOPY HELLO.* /S4

把所有与HELLO.*相相的文件，从用户 4 拷贝 到当前用户上去。

FILECOPY WHAT?.COM /D3S1

把所有与WHAT?.COM 相应的文件从用户 1 拷贝 到 用户了。

FILECOPY PIP.COM /S12D01

把文件PIP.COM 从用户12拷贝到用户 1。

一些出错信息

在 FILECOPY 操作过程中，可能 会出现下面一些出错信息：

↑C…aborted

当 FILECOPY 正 在 等 待 用户按一个键时，用户送入了 Control-c。程序将被废弃。

Failed to open SOURCE file correctly

当 FILECOPY 不能打开源文件时，显示出此信 息。这可能是由于读错，或插入了一个不正确的源盘所致。此时，程序也被废弃。

Failed to close DESTINATION disc correctly

当目的文件无法被关闭时，便产生这个信息。这可能是由

于磁盘的目录已满而所致。此时，程序将被废弃。

DESTINATION disc directory full

当在目的盘的目录中已没有空间再来建立一个新文件时，产生此信息。同时，程序被废弃。

DESTINATION disc full

当在目的盘中已没有更多的存储空间时，就产生出该信息。此时、程序也将被废弃。

The DESTINATION disc has an unknown format

当目的盘不具有三种AMSTRAD格式中的任何一个时，就产生此信息。如果磁盘来格式化，或磁盘已遭破坏，则也产生这信息。并反复显示目的盘的提示。

The SOURCE disc has an unknown format

如果源盘不具有三种AMSTRAD格式中的任何一种格式，如果盘末格式化，或磁盘已遭破坏，产生出这个信息。并反复显示源盘的提示。

SOURCE disc missing

已经按了一个键来响应对源盘的提示，但磁盘都不在驱动器中，则产生此信息。并反复提示。

DESTINATION disc missing

已经按了一个键来响应对目的盘的提示，但磁盘都不在驱

动器中，则产生此信息。并反复进行提示。

DESTINATION disc is write protected

当 FILECOPY 由于磁盘是写保护的而不能对目的盘进行写入时，显示该信息。此时，用户应该把去掉写保护片的目的盘再插进驱动器。目的盘的提示是反复显示的。

Incorrect DESTINATION disc

用户已经插进了一个目的盘，但这个磁盘与原来把文件拷贝到磁盘的盘不一样。只有当几个缓冲器的数据都要求拷贝一个文件时，才产生此信息。此时，程序被废弃。

Failed to read SOURCE disc correctly

如果源盘不完整，具有零的长度，或发生读错时，则产生此信息，且废弃掉程序。

Failed to write DESTINATION disc correctly

如果在对目的盘进行拷贝的过程中，发生了写错，则显示此信息。且程序被废弃。

WARNING：DESTINATION file <filename> is incomplete

如果 FILECOPY 程序不能成功地写入目的文件的<文件名>，则产生此信息。

中止信息

FILECOPY V2,1 abandoned

只要FILECOPY的正常操作受到影响时，便显示此信息。所拷贝的最后一个文件可能不完整。

FILECOPY V2,1 finished

当FILECOPY按照要求已经把所有的文件都成功地拷贝完时，显示此信息。

4。9 FORMAT。COM

介绍

FORMAT是一个CP/M的过渡程序，它的功能是对AMST-RAD CP/M的磁盘格式化。该程序支持三个AMSTRAD CP/M的磁盘格式。

FORMAT的起动

FORMAT程序存放在一个名为FORMAT。COM的文件中，并可利用命令：

FORMAT ＜可选的格式化参数＞

来调用这个程序。

FORMAT程序的参数

当调用FORMAT程序时，用户可以选取其所使用的格式，其方式是指定一种格式的参数。在FORMAT命令的后面，先安排至少一个空隔符，然后可以送入参数S，V，D或I四个参数中的任意一个。

S——系统格式

V——卖方格式

D——纯数据格式

I——IBMPC格式

S和V在磁盘上建立起同样的物理格式，其不同之处是：V参数对系统磁道不进行初始化。这就允许CP/M 程序被分配在CP/M磁盘上，而这个磁盘又不含有CP/M的操作系统。这样，就不致于违背Digital Research公司的许可协定。

如果不指定参数，则便假设选用S参数。如果用户送入一个非法参数（即不是S，V D或I）．那末，FORMAT程序就被废弃，且显示下面的信息：

Bad format option

(you may only enter S, V, D or I)

系统盘

当对系统盘格式化时，必须预光正确地建立起两个保留用的系统磁道。为此，FORMAT 在被调用时，它就对在驱动器A中的磁盘上读这两个磁道。如果该盘不是一个系统盘，则FORMAT程序废弃，显示下面信息：

The disc in drive Ais not a system disc

格式化

一旦格式参数有效，FORMAT程序便输出下面的提示：

Please insert the disc to be formated into

drive A then press any key.

当格式化开始及结束时，又显示出下列信息：

Formating started

Formating complete

从0磁道开始，依次对磁道进行格式化。在每一个磁道均格式化完之后，磁道上所有扇面都被读回，以确保它们的确已被正确地格式化了。而从每个扇区所读得的数据则不进行考虑。

多次格式化

当格式化完成之后，出现下面的提示：

Do you want to format another disc（Y/N）：

若想对另一个盘进行格式化，则用户应送入Y。若想退出FORMAT程序，并返回到CP/M，则用户应该送入N，结果，出现下面提示信息：

Please insert a cp/M system disc into
drive A then press any key：—

此时，用户应该在驱动器A中插入一个CP/M系统盘，并按一个键。FORMAT就开始热引导并返回CP/M。

其他一些信息

you must insert a CP/M system disc into
drive A

如果用户已被提示过需把一个CP/M系统盘插入驱动器A而没有这样去做，则输出该信息。显示此信息后，用户入再一次被提示要这样做。

you must insert the disc to be formated
into drive A

用户已被提示过要在驱动器A中插入一个盘片，而没有这样做。现再一次提示要这样做。

The disc to be formated in drive A must be write-enabled

驱动器A中的盘片是一个有写保护的盘片，现再一次提示用户，把一个目的盘插入驱动器A。

WARNING：Failed to format destination disc correctly
The destination disc should not be used until it is successdully formated

在格式化过程中，FORMAT程序被废弃时显示此信息。

The disc in drive A is not a CP/M system disc

FORMAT程序企图从一个非CP/M系统盘上读两个保留的系统磁道。该信息显示过后，程序被废弃。用户应该再一次在驱动器A中插进一个CP/M系统盘，并再一次调用此程序。

Formatting track：n

这里，n是当前的磁道号。当FORMAT程序对源盘正进行格式化时，显示此信息。

Failed to read source disc correctly：
track sector m

这里，n和m分别是坏扇区的磁道号和扇区号。如果FORMAT程序不能对保留的系统磁道的扇区正确读出时，显示此信息。程序也被废弃。

Failed to read destination disc correctly:

track n sector m

这里，n和m分别坏扇区的磁道号和扇区号。如果FORMAT程序不能正确地读出目的盘的一个扇区,该信息就被显示。且程序被废弃。

Failed to format destination disc correctly:

track n

这里，n是当前的磁道号。如果FORMAT程序不能对目的盘的一个磁道进行正确的格式化，则显示该信息。且程序被废弃。

↑C……aborted

当FORMAT程序正等待用户输入时，用户按了一个Control-C，则显示此信息。且程序被废弃。

Illegal message number: program aborted

在FORMAT程序执行过程中发生了错误。一般情况下,不应该如此。

4.10 SETUP.COM

介绍

SETUP是一个CP/M 的 过 渡程序，其功能是允许用户在一个AMSTRAD系统盘中的格式扇区中建立一些参数。

格式扇区

AMSTRAD CP/M允许用户在调用CP/M时建立某些参数

的始值。这些参数是存放在系统磁道上名为格式扇区的一个专用扇区中的。关于这方面的详细介绍可予阅节2．13。

SETUP是一个程序，它允用户设置一些参数值，这些参数值是存放在驱动器A的一个磁盘的格式扇区中。SETUP允许用户改变下列格式参数：

(1)初始命令缓冲器。

(2)起始字符串。

(3)打印机加电字符串。

(4)键盘转换。

(5)键盘扩充字符串。

(6)IOBYTE的设置。

(7)替换寄存器及IY寄存器的使能／禁止。

(8)BIOS信息的使能／禁止。

(9)初始命令缓冲器的清除。

(10)马达延时开启。

(11)马达延时关闭。

(12)驱动器步进速率。

(13)Z80 SIO通道A的时特率，数调位，校验位和停止位的数目。

(14)Z80 SIO通道B的时特率，数据位，校验位和停止位的数目。

SETUP的起动

SETUP程序存放在一个名为SETUP．COM的文件中，它可以用SETUP命令来调用，该程序通过格式扇区的所有的参数一步一步允许用户根据人们的要求来对参数进行改变。

如果格式扇区是无效的，那末 SETUP 便显示下面的信息：

Invalid config sector：

Do you want to use default setting（Y／N）：

如果用户送入N来响应上面的提示，则 SETUP 程序就废弃掉。否则，就需要用下列的约定参数：

（1）初始命令缓冲器为空。

（2）起始字符串为空。

（3）打印机加电字符串为空。

（4）不设置键盘转换。

（5）不建立键盘扩充字符串。

（6）约定的 IOBYTE 置成：CON：＝CRT：，RDR：＝TTY：，PUN＝TTY：和LST：＝LPT：

（7）替换寄存器和IY寄存器为使能状态。

（8）BIOS信息被使能。

（9）初始命令缓冲器将通过键盘输入而被清除。

（10）马达开启的延时设为50。

（11）马达关闭的延迟设为250。

（12）驱动器的步进速率为12毫秒。

（13）Z80 SIO通道A设为9600波特，8位数据位，无校验位，1位停止位。

（14）Z80 SIO通道B设为9600波特，8位数据位，无校验位，1位停止位。

下面几节详细说明 SETUP 程序是怎样允许用户来改变每一个参数的。

控制码的送入

大量的SETUP选择为文本输入提供提示。大多数的控制码可以利用键入相应的控制键而简单地送进去。但是，许多控制码具有专用的含意，而且不被送入到它本缓冲器的。这些控制码都列在下面。对于这些控制码的详细功能说明，可以参阅CP/M BDOS的功能10（读控制缓冲器）。

Control-C	在一行的开始时进行热引导。
Control-E	引起行的物理结束。
Control-H	退回一个字符位置。
Control-J	（换行），结束输入行。
Control-M	（回车），结束输入行。
Control-R	在新行之后再显示当前行。
Control-U	去掉当前行。
Control-X	与Control-U一样。

上述控制码（实际上是所有的其他代码）可以用一个上箭头后面跟着一个代表所要求的控制码的 ASCII 字符而进行送入。例如：↑C就认为是 Control-C，这个译码功能是通过把ASCII字符的最高三位置成0而实现的。因此，控制码也可以用三个ASCII 字符中的一个来表示，例如，Control-C 可以用↑#，↑C或↑c来表示。

两个上箭头后面跟了其他别的字符都被认作为是1。

初始命令缓冲器

这个缓冲器允许用户最多指令 128 个字符，这些字符在调用CP/M的时候将被自动地送入到CP/M中。如果这个缓冲器

是空的，则就显示下面的提示：

`•• Initial command buffer empty`

`Is this correct (Y/N):`

否则，SETUP程序就如下显示该缓冲器的当前内容：

`Initial command buffer:`

`"Command buffər"`

`Is this corret (Y/N):`

这里，"Command buffer"（命令缓冲器）是初始命令缓冲器的当前内容。如果用户送入一个Y响应上述提示，那末初始命令缓冲器就保持不变，并且SETUP移到下一个参数。如果用户送入一个N，则显示出下面的提示：

`Enter new initial command buffer:`

然后，用户可以送入该缓冲器所要求的不超过128个字符长度的参数。当新的缓冲器已被送入时，它就进行显示，并且用户被询问，它是否正确，如果不正确，那末这个过程一直重复到送入正确时才为止。

起始字符串

当调用CP/M时，这是一串输出到屏幕的字符。SETUP以与初始命令缓冲器中一样的方式来对用户进行提示。起始字符串最长可以是253个字符。

打印机加电字符串

当调用CP/M时，这是一串输出到BIOS列表设备的字符串。注意：这一个字符串送到哪一个实际的设备，则取决于IOBYTE的初始设置情况。SETUP以与初始命令缓冲器一样

的方式对用户进行提示，该字符串最长可以是253个字符。

键盘转换表

这一个表用来对键按下时所产生的代码进行重新定义。送到表中的每一个码有四个参数：

<键号>，<正常的>，<换档的>，和<控制的>

<键号>是一个范围为0到79之内的数，指明了哪一个键需重新定义。

<正常的>，，<换档的>和<控制的>这三个参数，如果有的话，都是一个范围为0到255内的数。这些参 数 对键按下时所产生的值进行重新定义。

<正常的>：指“换档”和“控制”都不按下时的值。

<换档的>：指只按下“换档”而不按“控制”时的值。

<控制的>是指只按下“控制”时的值。

由各个键所产生的值可解释如下：

0—31：具有专用功能的常用的一些控制字符。

32—127：普通的ASCⅡ字符。

128—159：根据所设置的扩充字符串所扩充的专用字符。

160—223：普通的字符——所有的CP/M 程序都不能在这个范围内对字符进行拷贝。

224—254：由固件为专用而保留。

255　　忽略。

如果不建立键盘的转换功能，则SETUP 便显 示下面的提示信息：

NO Keyboard translations set

Is this correctly (Y/N) :

否则，SETUP就显示如下的当前的键盘转换的设置：

键盘转换：

key code	Normal	shift	control
69	97	65	1
54	98	66	-

……etc。

Is this correct (Y/N) :

这里，“—”号意味转换维持不变。如果用户送入一个N来响应上面的提示，那末键盘的转换保持不动，并且使SETUP移到下一个参数。如果用户送入一个Y，则显示下面的提示：

Enter required command from :

A———Add key translation to table

D———Delete key translotion from table

C———Clear translation table

F———Finish translation

Command :

用户应该送入适当的命令字母，并在后面跟一个合适的参数的数。每个参数之间必须要用空隔符或逗号给予分开。由每一个命令所要求的参数如下所示：

A, <key number>, <nomal>, <shift>,
< control>

D, <key number>

C

F

注意：C和F命令不需要参数。

当F命令已被送入时，转换表就被显示，并且用户被询问是否正确，如果不正确，则此过程重复下去，直到正确时才为止。

键盘扩充表

这是一个重新定义32个扩充字符的表格。送入列表格的每一个内容都含有扩充字符以及与它有关的字符串。

通常情况下，规定键盘扩充0到12，这与数字键盘上的键所相应。键盘扩充表允许用户加入附加的扩充，或对规定的扩充进行修改。

如果没有额外或修改的键盘扩充被置，则SETUP就显示下面的提示信息：

NO keyboard expansions set

Is this correct（Y/N）：

否则，SETUP显示如下的额外的或修改的键盘扩充：

Keyboard expansion strings：

Expansion Token	Expansion string
0	DIR↑M
1	STAT↑M

etc……

Is this correct（Y/N）：

如果用户送入一个Y作为响应，则键盘扩充将保持不变，而且使SETUP移到下一个参数。如果用户送入一个N，则显示下面的信息：

Enter required command from：

A——Add keyboard expansion to table

D——Delete keyboard expansion from table

C——Clear expansion table

F——Finish keyboard expansions

Command:

然后，用户应该送入相应的命令字母，它的后面再跟一些合适的参数。每个参数之间一定要用空隔符或逗号来分开。每一个命令所要求的参数如下：

A，<key number>，<expansion string>

D，<key number>

C

F

注意：C命令和F命令不要求有参数。A命令要求扩充字符串为一个不超过30个字符的长度。当F命令被送入时，扩充表就显示出来，并且询问用户是否正确，如果不正确，则反复进行这一过程，直到正确时才为止。

设置IOBYTE的约定值

约定的IOBYTE的设置是在调用CP/M时，送到CP/M IOBYTE的位于RAM的#0003地址中的数。SETUP显示了IOBYTE的当前的设置情况：

Default IO byte settings are:

CON: is assigned to CON-DEV.

RDR: is assigned to RDR-DEV.

PUN: is assigned to PUN-DEV.

LST：is assigned to LST-DEV.

Is this correct（Y/N）：

这里，

CON-DEV是对CON：所指定的约定设备。

RDR-DEV是对RDR：所指定的约定设备。

PUN-DEV是对PUV：所指定的约定设备。

LST-DEV是对LST：所指定的约定设备。

如果用户送入Y来响应上述的提示，那末约定的IOBYTE设置便保持不变，并且SETUP移到下一个参数。如果一个用户送入了N，那末便显示下面的提示：

Enter required IO byte setting：

用户然后可以指定4个CP/M的逻辑设备中的一个给与它有联系的物理设备中的一个设备。

<逻辑设备><物理设备>

这里，<逻辑设备>是CON：，RDR：，PUN：，LST：之一，而<物理设备>则是CRT：，TTY：，BAT：，UCI：，PTR：，UR1：，UP2，UP1，UP2，LPT：，UL1中的一个。两个设备之间必须用一个空隔符，逗号或等号来分开。例如CON：=CRT：表示了把逻辑设备CON：分配给物理设备CRT：。详细地讲，所有的逻辑设备分配给物理设备的有效情况如下：

CON：可以分配给：TTY：CRT：BAT：UC1：

RDR：可以分配给：TTY：PTR：UR1：UR2：

PUN：可以分配给：TTY：PTR：UP1：UP2：

LST：可以分配给：TTY：CRT：LPT：UL1：

上述分配的更详细说明可参阅2.7节。

当已经送入一个新的约定的IOBYTE设置之后，它就进行

显示，并且询问用户是否正确，如果不正确，则重复上述过程，直到正确时为止。

替换寄存器和IY寄存器保留的使能/禁止（方式）

方式设置确定了BIOS保留还是不保留替换寄存器和IY寄存器。SLOW方式表示了BIOS保留这些寄存器，并且是一种被推荐使用的方式。FAST方式表示了BIOS不保留这些寄存器。术语SLOW和FAST是一种不确切的说法。SETUP提示如下：

Default mode setting is <方式>

Is this correct（Y/N）：

这里，<方式>是约定的方式，要么是fast，要么是slow。如果用户送入Y作响应，则原先的设置保持不变，且SETUP移向下一个参数。如果用户送入N，则原先的方式就改变到其他状态，且询问用户是否正确，若为不正确，则反复上述过程，一直到正确时为止。

除了有某种特殊要求以外，用户最好不要选用FAST方式。如果要求使用替换寄存器或IY寄存器的应用程序都不会在FAST方式中工作，例如：DRLOGO程序。

BIOS的信息的使能/禁止

该参数值确定了在调用CP/M时，BIOS的出错信息要不要显示出来。SETUP显示当前的约定状态如下：

Default：BIOS message <状态>

Is this correct（Y/N）：

这里，<状态>是约定的信息状态，要么使能，要么禁

止。如果用户用Y来作响应，则原先的对信息的设置保持不变，且SETUP移到下一个参数。如果用户送入N，则原先的状态变成其他状态，并询问用户是否正确，若不正确，则上述过程一直反复进行到正确为止。

初始命令缓冲器的清除/保存

该参数的值确定了初始命令缓冲器器是否进行清除（当在键盘上按下一个键的时候）。SETUP显示了如下的当前约定状态：

Default：<状态> initial command buffer on keyboard input

Is this correct（Y/N）：

这里的<状态>指保存还是清除。如果用户用Y来作出响应，那么原先的状态保持不变，且SETUP移向下一个参数。如果用户N作响应，那么原先的状态就改变成其他状态，且询问用户是否正确，若为不正确，则一直反复进行到正确时为止。

驱动器马达开启延时

马达的开启延时规定了马达被开启后一直到对磁盘进行存取一般时间里BIOS所必须处于等待的时间的长短。该时间是以1/50秒为单位而表示的。SETUP显示了马达开启延时的当前状态为如下：

Default motor on delay is <nn>1/50 second units

Is this correct（Y/N）：

这里，<nn>是当前的以 1/50 秒为周期所测得的马达开启延时时间。如果用户用Y来作响应，则马达开启延时时间保持不变，且 SETUP 移向下一个参数。如果用户送入一个 N，则显示出下面的提示：

Enter new motor on delay in 1/50 second units：

为此，用户可以送入一个所需要的马达开启延时值。当新的值已被送入后，它就会显示出来，并询问用户是否正确。如果正确，则一直进行到正确时为止。

对DDI-1/FD的推荐值为 1 秒，也就是 50。更大一些的数值就会毫无必要地降低磁盘系统的速度，而更小的数会使磁盘发生假错。

驱动器马达的开关延时

马达关闭延时时间系指最后一次对磁盘进行存取到马达开关时，BIOS所必须要有的等待时间。该时间也以 1/50为单位。SETUP显示了马达关闭延时的当前设置为：

Default motor off delay is <nn>1/50 second units

Is this correct（Y/N）：

这里，<nn>是以1/50秒为周期而测量的当前的马达的关闭延时。如果用户以Y作出响应，则马达关闭延时保持不变，且SETUP移向下一个参数。如果用户送入了 N，则显示下面的提示：

Enter new motor off delay in I/50 second units：

此时，用户可以送入马达关闭延时的要求值。当新的值已经送入后，它就会显示出来，并询问用户是否正确，如果不正确，则上述过程反复进行到正确时才为止。

所推荐的值为5秒，即为250。在一个磁盘进行操作以后再关闭马达其理由是为了避免另一个磁盘操作在这之后很快便开始操作的情况下的马达起动延时。但是，马达关闭的时间太长，会引起驱动器所不必要的减慢。5秒的时间是一个折衷的数值，如果需要的话，用户还可以随意地增大这个时间。

步进速率

步进速率规定了磁盘驱动头可以在磁盘上步进的速度。其时间是用毫秒来指明的。SETUP 显示了步进速率的当前设置如下：

Default stepping rate is <nn> milliseconds

Is this correct（Y/N）：

这里，<nn>是以毫秒为测量单位的当前的步进速率。如果用户以Y作出响应，则步进速率保持不变，且SETUP移向下一个参数。如果用户送入N，则显示下列提示：

Enter new stepping rate in millisecond：

为此，用户可以送入一个所要求的步进速率值。当新的数值送入之后，它被显示出来，并询问用户是否正确，如为不正确，则一直反复进行到正确时才为止。

对DDI-1/FD-1驱动器来讲，步进速率主为12毫秒。其他的驱动器可能要求有不同的步进速率。

串行接口通道A的格式

对于串行接口的通道A的格式，可以指定许多参数。这些参数有：

Tx波特率——发送波特率的约定值。

Rx波特率——接收波特率的约定值。

数据位——数据位的约定的数目。

校验位——校验设置情况的约定。

停止位——停止位的约定的数目。

SETUP显示的Z80SIO通道A的约定设置情况如下：

```
Z80 SIO Channel A: <AA> tx baudrate,
<BB> rx baudrate, <CC> data bits,
<DD> parity, <EE> stop bits
  Is this correct (Y/N):
```

这里，<AA>是规定的发送波特率。

<BB>是规定的接收波特率。

<CC>是规定的数据位长度，5，6，7或8。

<DD>是规定的校验设置，偶，奇或无。

<EE>是规定的停止位位数，1，1.5或2。

系统支持有下列的波特率：

19200　9600　4800　3600　2400　2000
1800　1200　600　300　200　150
110　75　50

如果用户送入Y作为响应，则通道A的约定格式保持不变，且SETUP移向下一个参数。如果用户选入N，则显示出下面的提示：

```
Enter Channel A Parameters:
```

此时，用户可以送入所要求的格式参数，并且要以上面规定的次序送入。如果任何一个参数为无效，或超过了范围，则便输出一个相应的出错信息，而用户必须再一次送入全部参数。当新的数值已经成功地送入，则它们会被显示出来，且用

户被询问是否正确，如果不正确，则重复进行到正确时才为止。

串行接口通道B 的格式

对于串行接口扩充板的通道B的格式，可以规定许多的参数。这些参数有：

波特率——发送及接收波特率的约定值。

数据位——数据位的约定位数。

奇偶——奇偶位的约定设置。

停止位——停止位的约定位数。

SETUP以如下格式显示了Z80 SIO 通道B的约定设置情况：

Z80 SIO channel B：<AA> baudrate，<BB>
bata bits，<CC> parity，<DD> stop bits
Is this correct（Y/N）：

这里，<AA>是约定的发送及接收波特率。

<BB>是约定的数据位位数。5，6，7或8。

<CC>是约定的奇偶设置。偶、奇或无。

<DD>是约定的停止位位数。1，1.5者2。

系统支持有下列的波特率：

19200	9600	4800	3600	2400	1200
1800	1200	600	300	200	150
110	75	50			

如果用户Y来响应上面的提示，则通道B的原先格式保持不变，并且，所有的参数全部设置完毕。如果用户送入N，则显示出下列提示符：

Enter channel B parameters：

此时，用户可以用上面所规定的次序送入所要求的格式参数。如任何一个参数为无效，或超出参数应选的范围，那么就输出一个相应的出错信息，用户就必须重新送入所有的参数。当新的参数值已经送毕，它们就被显示，且询问用户是否正确，如果不正确，则反复进行上述过程，一直到正确时为止。

系统盘的修改

一旦用户已经建立好全部参数之后，SETUP 便发生如下的提示：

Do you want to update your system disc（Y/N）：

如果用户送入N，则 SETUP 便被废弃，而且对系统盘不作修改。如果用户送入Y，则驱动器A中的磁盘用新设置的参数系统修改，然后再显示下面的提示信息：

Do you want to restart CP/M（Y/N）：

如果用户送入N，则 SETUP 便通过一个热引导而退出。如果用户送了Y，则SETUP便启动一个CP/M的完整的加电初始化程序，这样就使最新的格式过的参数开始起作用。

SETUP程序的一些信息

SETUP程序含有下列一些信息：

Failed to read configuration sector correctly

SETUP不能正确地读出格式扇区。

输出该信息之后，SETUP程序被废弃了。

Failed to write configuration sector correctly

SETUP不能正确地对格式扇区进行写入。

输出该信息之后，SETUP程序便废弃了。

Failed to verify configuratioo sector correctty

从格式扇区读回的数据与对它所写入的数据不相等。

输出该信息之后，SETUP程序就被废弃了。

Sector buffer overflow

在格式扇区里，没有足够多的空间来存放缓冲器、字符串和表格。

输出了该信息之后，SETUP 程 序就被废弃掉，并且对格式扇区不作修改。

The disc in drive A is not a system disc

SETUP企图从一个非系统盘中读格式扇区。

该信息显示之后，SETUP程序就被废弃了。

Bad control character

用户输入了一个上箭头（↑）作为最后一 个 字 符 的字符串。

Bad command option （Enter A，D，C或F）

在设置键盘转换或扩充缓冲器时，用户送入了一个无效的可选命令。

Invalid number input

SETUP 正指望输入一个有效数的时候，但是没有查找到或没有用分界符来作为结束符。（分界符可用空隔符、逗号或行结束符）。

keyboard translation table empty

在没有建立转换的情况下，用户想在键盘转换表中删除掉一个入口。

keyboard translation table full

当转换表已经都满了的时候，用户还想在键盘转换表中加进一个入口。注意，系统实际上已经准备了可保存最多为80个入口的空间，所以这个出错信息实际上是不可能会产生的。

key numbers must be in the range 0 to 79

用户所指定的键不在0到79的范围之内。

key codes must be in the range 0 to 255

用户所指定的一个键转换码不在0到255范围内。

keyboard expansion buffer empty

在没有建立扩充时，用户想在键盘扩充缓冲器中删去一个入口。

keyboard expansion buffer full

键盘扩充缓冲器溢出。

key tokens must be in the range 0 to 31

用户规定的键盘扩充标志不在0到31范围内。

Invalid command
(you must only specify CON：，RDR：，PUN：
or Lst：)
用户指明了一个非法的IOBYTE逻设备。

CON：may only be assigned to
TTY：，CRT：，BAT：，or UC1：
用户企图将CON：分配到一个非法的物理设备。

RDR：may only be assignea to
TTY：，PTR：，UR1：，or UR2：
用户企图将RDR：分配到一个非法的物理设备。

PUN：may only be assigned to
TTY：，PTP：，UP1，or UP2：
用户企图将PUN：分配到一个非法的物理设备。

LST(*)：may only be assigned to
TTY：，CRT：，LPT：or UL1：
用户企图将LST：分配到一个非法的物理设备。
(*)原文LST：误印为CON：。　　　　　　译者注

Step rate must be in the range 1 to 32
用户所规定的步进速率不在范围1到32之间。

Invalid baudrate (you must only specify)
19200, 9600, 4800, 2400, 2000, 1800, 1200,
600, 300, 200 150, 110, 75 or 50)
用户对两个SIO口中的一个规定了一个非法的波特率。

Invalid parity
(you must only specify ODD, EVEN or NONE)
用户对两个SIO口中的一个口规定了非法的奇偶设置。

Invalid data bits
(you must only specify 5, 6, 7or8)
用户对两个SIO 口中的一个口规定了一个非法的数据位位数。

Invalid stop bits
(you must only specify 1, 1.5 or 2)
用户对两个SIO 口中的一个口规定了一个非法的停止位的位数。

↑C……aborted
当SETUP 程序 E 在为用户输入而处于等待时，用户送入了一个Control-c。
该信息输出之后，SETUP程序就被废弃了。

Illegal message number: Program aborted
该信息表示了在执行 SETUP 程序时发生了一个错误。一

般情况下不应该出现这个信息。

4.11 BOOTGN.COM和SYSGEN.COM

介绍

所提供的BOOTGEN和SYSGEN程序允许在系统盘上对系统磁道重新格式化，或在用户盘上建立一些系统磁道。系统道的重新格式化可以通过分配一个非标准尺寸的CP/M 系统（用MOVCPM来完成），或分配一个非标准格式（利用SETUP来完成）而实现。

用户盘到系统盘的转换

用户盘都是一些具有空白的系统道的CP/M 系统盘。它们主要是用于分配一些应用程序的软件，并且在使用它们之前需要加上一些系统道的信息。

利用 FILECOPY 程序，把一个用户盘转换成一个可适宜的系统盘是可行的，这样做，避免了反复使用SYSGEN或BOOTGEN的必要性。但是必须小心，不要把原始的用户盘改变掉。不过，可以就它的复制盘进行变换（使用 DISCCOPY 或COPYDISC）。

转换工作是通过对每一个实用程序进行调用而完成的。

调用BOOTGEN时，首先应该简单地送入一个“BOOTGEN”的命令。

程序首先对源盘请求，从源盘上将读出引导程序以及格式信息——正确地键入〔ENTER〕以便使用已经位于驱动器中的系统盘，要不然就插入一个所需要的替换系统盘，然后再键入〔ENTER〕。

当信息已被读出时，BOOTGEN就对目的盘进行请求，此时，你应该在驱动器中插入一个你所希望进行转换的用户盘。

在程序结束时，当对一个CP/M 盘提出请求时，你必须插入一个正确的系统盘，因为在这个盘上你只写入了所要求的CP/M信息的一半的内容。

为了生成另一半CP/M 的信息，则必须同样地再严格地执行一次SYSGEN。

当从第二个实用程序中退出时，用户盘便有了和系统盘同样的格式了。

非标准版本的CP/M的种类

系统磁道的信息是放在一个系统盘的保留的一些磁道上的两个不同的块中。一部分含有引导段(即在起动时需执行什么)和格式段(它含有了象开始信息、键转换和磁盘驱动器硬件参数等一些信息）引导段在所有的系统盘都是一样的，即使所构成的CP/M 的尺寸不一样，但它们执行什么功能都都是不变的。格式段则对某一特定的应用是独特的，并且是独立设置的。因此，这些段在一个已存在的磁盘上要改变CP/M 的尺寸时，不需作修改。

第二部分中的系统数据包含了CCP 和 BIOS，对于不同尺寸的CP/M，它们都需要重新格式化过，对于一个不同尺寸的CP/M来说，CCP和BIOS是通过MOVCPM 来建立的，并且当程序执行完毕后，它们是留在存储器中的(见2．6)。SYSGEN可以通过这个存储器的映象直接重新生成系统磁道，也可以通过一个文件对它进行复制而重新生成系统磁道。这样的一个存储器的文件复制是通过调用“SAVE 34…”，由MOYCPM 作

为提示而完成的。

SYSGEN程序也做了某些校验工作，以保证写入到系统磁道上的有关内容是正确的。

为了重新分配好系统磁道，SYSGEN应该用下面两种格式中的一个来进行调用：

SYSGEN*

SYSGEN<文件名>

SYSGEN*在一个MOVCPM<nnn>*下，由存储器映象来生成系统磁道。这里，<nnn>是CP/M系统的尺寸。（见2．6节。注意，参数*对于MOVCPM是一定要有的。

SYSGEN<文件名>在指定的文件中，通过有关的记录而生成了磁道。<文件名>中不得含有任何的通配符。

非标准格式的类别

SETUP程序提供了一种复制格式扇区的方法。利用运行SETUP而对所要求的目的盘的结果保存并不改变的手段，有可能将格式区从一个磁盘转换到另一个磁盘上。而更为简单的是运行一个BOOTGEN程序，它将对引导段进行复制（引导段在所有的磁盘上是一样的），并复制所要求的格式段。该程序将提示出源盘和目的盘。

概括

BOOTGEN　　复制引导段和格式段。

SYSGEN　　复制CCP和BDOS

SYSGEN*　　从TPA映象中保存CCP和BDOS

SYSGEN<文件名>　　从TPA映象所保存的文件中写入

CCP和BDOS。

BOOTGEN和SYSGEN的一些信息

Please insert SOURCE disc into drive A
and press any key
Please insert DESTINATION disc into drive A
and press any key

当程序要求对一个盘进行读一个文件时，或对系统道要进行读或写的时候，产生上面的信息。

File not found

已在调用行中所指定的源文件（一般是由一个MOVCPM来产生的）不能在所插入的盘上的当前所选择的用户号中找到。

Loading

表示在调用行中所指定的源文件已经在磁盘上找到并企图装入有关的部件。

<文件名>has incorrect format

所指定的源文件没有足够多的记录来存放所要求的内容。

Ambiguou SOURCE filename

所指定文件名含有通配符，因此无法正确打开。

↑C……aborted

在程序执行期间，用户按下了 Control-C，因此，操作被中止。

MOVCPM memory image not present in TPA

用户正企图在未从MOVCPM 所建立的数据，或不是由现有的系统道上取得的数据中生成系统磁道。

SOURCE disc is not system farmat

DESTINATION disc is not system format

用户试图在一个盘上利不同于系统的格式来对系统道进行读/写，因此，不能占有系统道。

SOURCE disc has unknown format

DESTINATION disc has unknown format

用户试图对一个没有标准的AMSTRAD格式的盘进行读／写。

SOURCE disc missing

DESTINATION disc missing

程序正试图在驱动器没有盘的情况下执行一种磁盘操作。

DESTINATION disc is write protected

由于磁盘具有写保护，程序因而不能对它进行写入。

SOURCE disc error

DESTINATION disc error

由于上述所说明的情况以外的情况，而引起的读／写出错。

在一般的程序执行过程中，不该有这个信息。

Do you wish to reconfigure

another disc（Y／N）？：

当目的盘的系统道已正确写入时，产生此信息。注意：任何其他磁盘均应使用与首次相同的系统道内容。若要使用不同的内容，程序将需要重新执行。

Please insert a CP/M disc in drive A then press any key：

当程序圆满地退出，或中止时，产生此信息。表示了应该再插入一个可以进行热导的CP/M盘。

第五章　硬　件

本章说明了在DDI-1 中的磁盘接口在使用其他一些驱动器时的一些注意事项，并对串接口的硬件构成作一个较详细的介绍。

注意的是，DDI-1中不提供串接口的。

5．1　磁盘接口

软盘控制器

该控制器使用的是NEC的型号为 μPD765A 软盘控制器集成电路与磁盘驱动器相连接。仅仅只提供了两个磁盘驱动器，这是因为 μPD765A中的US1 线被不考虑之原因。这两个驱动器作为驱动器 0 和 1，又可作为驱动器 2 和 3 来存取。控制器支持单面的和双面的以及单密度和双密度的小型软盘驱动器。注意：对于 μPD765A的CLK引腿所提供的时钟频率为 4MHZ，而不是大型磁盘驱动器所用的8MHZ时钟频率。

对于 μPD765A来讲，凡NEC 的数据表中所介绍的全部特性都可采用，但不支持中断和DMA。

磁盘接口使用Z80 I/O的口如下：

口	输出	输入
#FA7E	马达控制	**不用**

#FB7E	**不用**	μPD765A状态寄存器
#FB7F	μPD765A数据寄存器	μPD765A数据寄存器

扩充ROM

磁盘的扩充ROM位于接口板上。通常该ROM编号为#07，但也可以利用把LK1线割断的方法，将其编号置为#00。当ROM的编号为#07时，需将板上的EXP信号线（引腿48）接地，这样，可以避免该地址被其他扩充的外设去使用。一个200微毫秒的27128EPROM或ROM通常被使用，并且可以装在板上的DIL（双列直插型）插座上。

跳接点LK2和LK3是在出厂时选接好，其作用是进行写予补偿用。它们不应该由用户来改变。

马达的控制

写入到这个通道的数据可用来起动或停止磁盘驱动器的马达。写入#00表示停止马达转动，写入#01表示启动马达。在加电或其他系统复位时，马达均不停止。

联接插头的型号

装在本单元上的联接插头是一根排列好的装在印制板上的34线扁平电缆连接插头座。

电平

在控制器中，所有电平都与TTL相兼容。在驱动器这端所发出的信号用与$+5_v$相接的680欧姆的电阻来终结，在控制器及接收端的信号则用一个具有输入延滞的门电路与其相接。联

接电缆的最大容许长度为0.75米。

引腿的安排

所有奇数引腿均接地，所有信号都是低电平作用。

引腿号	信号
2	$+5^{V}$
4	$+5^{V}$
6	$+5^{V}$
8	索引
10	驱动器送选 0
12	驱动器送选 1
14	$+5^{V}$
16	马达开启
18	方向选择
20	步进
22	写 数 据
24	写 选 通
26	磁 道 0
28	写 保 护
30	读 数 据
32	面 1 选择
34	准 备 好

其他磁盘驱动器的使用

对于DDI-1来讲，使用另一种磁盘驱动器，特别是$5\frac{1}{4}$英

寸的驱动器是可能的。这时，需要另外一些硬件知识。下面提供一些注意事项，及有助于使用不同的驱动器。

对于驱动器A：在确保任何联到驱动器的线路都已去掉之后，将+5ᵛ电源接到5¼英寸驱动器的引腿2，4，6和14上。

对于驱动器B：不应该装终端电阻。

驱动器必须要有一个“准备好”信号（引腿34）。

驱动器应该有它本身的电源。

附加的电缆应该尽可能地短，并应该有一个装好线缆的公插头（以与接口处的母插座相联），一般来讲，它是一个34线的双面板的边缘联接插头（以与5¼英寸驱动器相联）。

步进速率、马达启动和停止越时，都可以改变。（见SETUP程序）。

但是，象Shugart201这样的驱动器是不宜使用的，因为它没有准备信号。但是象Chinon FO51-MD这样的驱动器是可以使用的。

5.2 串行接口

BIOS支持有一个双通道的异步串行接口。但是，这样一个接口不是DDI-1的一部分。这个串行接口如果配置了的话，则可以由BIOS来驱动它。

这个接口是由一个Zilog Z80A SI0／0或Z80A DART与一个Intel 8253可编程间隔定时器所组成。8253被用作为波特率发生器。

定时器0：产生SIO的通道A的发送时钟。

定时器1：产生SIO的通道A的接收时钟。

定时器2：产生SIO的通道B的发送和接收时钟。

假定，这个器件的所有三个通道的CLK（时钟）输出都是由一个2.0（±0.1%）MHZ的时钟信号所驱动，这个2.0 MHZ的时钟信号则又是通过4.0MHZ的CPU时钟而引出的。所有三个通道的GATE输入端都被连在一起，且始终为高电位。

I/O口如如下所使用：

口	输出	输入
#FADC	SIO通道A数据	SIO通道A数据
#FADD	SIO通道A控制	SIO通道A控制
#FADE	SIO通道B数据	SIO通道B数据
#FADF	SIO通道B控制	SIO通道B控制
#FBDC	8253装计数器0	8253读计数器0
#FBDD	8253装计数器1	8253读计数器1
#FBDE	8253装计数器2	8253读计数器2
#FBDF	8253写方式字	**不用**





